IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2016, No. 2, pp. 288–306. DOI:10.13154/tosc.v2016.i2.288-306

Multi-key Analysis of Tweakable Even-Mansour with Applications to Minalpher and OPP

Zhiyuan Guo[1,2,4], Wenling Wu[1,2,4], Renzhang Liu[3] and Liting Zhang[1]

[1] TCA Labaratory, State Key Laboratory of Computer Science (SKLCS), Institute of Software, Chinese Academy of Sciences, Beijing, China
gzhyuan@msn.cn,wwl@tca.iscas.ac.cn,liting.zhang@hotmail.com
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
liurenzhang@iie.ac.cn
[4] University of Chinese Academy of Sciences, Beijing, China

Abstract. The tweakable Even-Mansour construction generalizes the conventional Even-Mansour scheme through replacing round keys by strings derived from a master key and a tweak. Besides providing plenty of inherent variability, such a design builds a tweakable block cipher from some lower level primitive. In the present paper, we evaluate the multi-key security of TEM-1, one of the most commonly used one-round tweakable Even-Mansour schemes (formally introduced at CRYPTO 2015), which is constructed from a single n-bit permutation P and a function $f(k,t)$ linear in k from some tweak space to $\{0,1\}^n$. Based on giant component theorem in random graph theory, we propose a collision-based multi-key attack on TEM-1 in the known-plaintext setting. Furthermore, inspired by the methodology of Fouque et al. presented at ASIACRYPT 2014, we devise a novel way of detecting collisions and eventually obtain a memory-efficient multi-key attack in the adaptive chosen-plaintext setting.

As important applications, we utilize our techniques to analyze the authenticated encryption algorithms Minalpher (a second-round candidate of CAESAR) and OPP (proposed at EUROCRYPT 2016) in the multi-key setting. We describe known-plaintext attacks on Minalpher and OPP without nonce misuse, which enable us to recover almost all $O(2^{n/3})$ independent masks by making $O(2^{n/3})$ queries per key and costing $O(2^{2n/3})$ memory overall. After defining appropriate iterated functions and accordingly changing the mode of creating chains, we improve the basic blockwise-adaptive chosen-plaintext attack to make it also applicable for the nonce-respecting setting.

While our attacks do not contradict the security proofs of Minalpher and OPP in the classical setting, nor pose an immediate threat to their uses, our results demonstrate their security margins in the multi-user setting should be carefully considered. We emphasize this is the very first third-party analysis on Minalpher and OPP.

Keywords: Multi-key Setting · Tweakable Even-Mansour Scheme · Authenticated Encryption · Collision-based Cryptanalysis · Minalpher · OPP

1 Introduction

1.1 Multi-key Analysis

With regard to the cryptanalysis, cryptosystems are mostly evaluated in the single-key and related-key models. In the former, adversaries have access to the scheme equipped

Received: 2016-09-01, Accepted: 2016-11-01, Published: 2017-02-03

with a uniformly random key, without any knowledge of the key. In the latter, the scheme is equipped individually with related keys, whose values are secret but relations are known. Both models have shown great benefits in analyzing cryptographic schemes, and continuously motivated the advance of practically secure new designs [BW00, BK09, DFJ13, Mav15]. However, even when the schemes show sufficient strength in the above two models, in practical applications their secret keys need to be renewed within every particular period (usually called key lifetime), to avoid key guessing attacks by brute force and keep sufficient security margin against normal attacks. Furthermore, the renewed keys should be random as well as independent from all previously used keys.

As a consequence, a new model that assumes multiple independent keys are equipped individually in the same cryptosystem has more practical significance. Mantin et al. [MS01] capture this by proposing broadcast setting, where a single unknown plaintext is encrypted for several times with distinct keys and then sent to individual recipients. Afterwards, Chatterjee et al. introduce multi-user setting [CMS11], in which the same message is either authenticated or encrypted with multiple keys, and describe how the adversary can recover one key by observing tag or ciphertext collisions and corrupting its related key. Recently, Mouha et al. formally define the multi-key setting [ML15], a more generalized model where plaintexts need not be the same in communications to different users, and the secret keys need not be corresponding to distinct users. All the same, the distinct keys should be uniformly random and independent from each other. Adversaries can collect scheme outputs (corresponding to their chosen inputs or not) under all keys, and try to corrupt the scheme security under any of them.

Obviously, the multi-key setting is more close to practice than the broadcast and multi-user setting (also than the usual single-key/related key model). The reason is that even for a single user, she may encrypt or authenticate plaintexts with multiple keys due to the frequently happened re-keying operations. Such operation is often resulted by the common implementation practice to use session keys [MOV96], and is also necessary in certain scenarios to avoid cryptographic attacks or to comply with existing standards [BDJR97]. In practical communications, IETF requires that IKE, ESP, and AH security associations use secret keys that should be used only for a limited amount of time [RFC05]. NIST not only recommends to limit the number of message blocks under the same key, but also to limit the number of MAC failures before rekeying is required [Dwo05]. For example, the amount of plaintext that can be processed under the same key is limited to 32 GB for 3-key Triple-DES, and to 16 MB for 2-key Triple-DES [BB12]. Any GCM key that is established among its intended users shall, with high probability, be fresh [Dwo07].

More specifically, multi-key analysis has shown strong ability in evaluating practical schemes. AlFardan et al. [ABP+13] show that it is a realistic attack vector in the case of TLS to obtain the encryption of one secret under multiple independent keys. They explain that this can be done by either using JavaScript malware to generate multiple sessions, or causing the session to be terminated, after which some applications automatically reconnect and retransmit the cookie or password. Paterson et al. mount an effective statistical plaintext recovering attack on RC4 in IEEE WPA/TKIP [PPS14], and further improve the attacks by recovering user passwords in TLS [GPV15]. In a nutshell, just as insisted by Menezes in the invited talk [Men12] at EUROCRYPT 2012, cryptographers have to consider multi-key analysis when devising or analyzing cryptosystems.

1.2 Tweakable Even-Mansour

Given a public permutation P, the simplest way to design a block cipher is to iterate P by inserting random keys $k_1, \cdots, k_{r+1}$ in between, i.e.

$$EM_{k_1,k_2,\cdots,k_{r+1}}(m) = P(\cdots P(P(m \oplus k_1) \oplus k_2) \oplus \cdots) \oplus k_{r+1},$$

where the round keys can be either independent of each other or derived from a master key. One-round such construction is firstly analyzed by Even et al. [EM97], and r-round such construction is specially introduced as key-alternating ciphers to facilitate the analysis on AES [DR02], which inspires many new designs of symmetric cryptographic algorithms. We interchangeably call $EM_{k_1,\cdots,k_{r+1}}$ an Even-Mansour cipher. Its distinguishing features, e.g. simple and elegant structure, light or even no key scheduling, make it extremely suitable for compact implementations and thus attract considerable attention in recent years [Din15, DKS12, FJM14, ML15].

In addition, tweaking such ciphers with an extra input is an interesting and useful direction, in the spirit of tweakable block ciphers proposed by Liskov et al. [LRW11]. By sharing a single key with each other, tweakable block ciphers behave like independently (super) pseudorandom permutations, as long as their tweaks are pairwise distinct. The obtained variability over standard block ciphers and saved re-keying operations greatly simplify the design work for modes of operation [Rog04].

As early approaches for tweakable Even-Mansour construction, Sasaki et al. propose a concrete one-round scheme and use it to design their AE algorithm Minalpher [STA+15], which is a second-round candidate in the CAESAR competition [CAE14]. Besides, Jean et al. present a general construction named TWEAKEY framework, adopting tweaks in key scheduling of EM ciphers [JNP14]. Then independently, Cogliati et al. [CS15b] and Farshim et al. [FP15] consider a simple tweak method $k \oplus t$ in each round of EM cipher, and find attacks against two rounds and birthday-bound security for three rounds. Formally, Cogliati et al. introduce the following one-round construction:

$$TEM(k,t,m) = h_k(t) \oplus P(h_k(t) \oplus m),$$

where h is a uniform and almost XOR-universal family of hash functions and t is a tweak from some space [CLS15]. In the random permutation model, they show that its 2-round variant is secure up to $2^{2n/3}$ and r-round approaches optimal security when r grows. However, since the non-linearity of h will inevitably slow down the efficiency, they next aim at beyond-birthday-bound security by use of only linear tweak and key mixing, and succeed with four rounds [CS15a]. Most recently, Granger et al. revisit this approach through improving masking methods with word-oriented LFSR and powering-up-based techniques [GJMN16]. Their proposed MEM construction is finally instanced as new AE schemes OPP and MRO, and shown with competitive speed.

1.3 Our Contribution

In the present paper, we evaluate the multi-key security of TEM-1, a one-round tweakable Even-Mansour scheme shown in Figure 1, which can be characterized as:

$$TEM(k,t,m) = f(k,t) \oplus P(f(k,t) \oplus m),$$

where P is an n-bit public permutation, k is a secret key, t is a tweak, and $f(k,t)$ is a function linear in k. As discussed by [FJM14, ML15], the attack in multi-user/key setting should be more efficient than independent executions of the attack in single-key setting. Namely, in order to recover all independent keys in a large set of size L, the time complexity

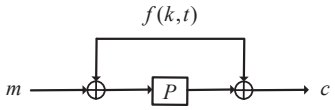

Figure 1: TEM-1: One-round Tweakable Even-Mansour with function $f(k,t)$ linear in k

of multi-key analysis should be no more than $L \cdot 2^{|k|}$, where $|k|$ denotes the length of a single key.

With this goal in mind, we propose effective multi-key analyses on TEM-1 in different attack models. Several related techniques used in the collision-based cryptanalysis are first introduced in Section 2. With the help of giant component theorem in random graph theory, we afterwards present known-plaintext attacks against TEM-1 in Section 3, which enable us to recover almost all $O(2^{n/3})$ independent keys by making $O(2^{n/3})$ queries per key and costing $O(2^{2n/3})$ memory overall. Furthermore, inspired by the idea of Fouque, Joux and Mavromati, we in Section 4 devise a novel way of detecting collisions to obtain an adaptive chosen-plaintext attack, which remarkably reduce the memory complexity to $O(2^{n/3})$.

As important applications, Section 5 directly utilizes our techniques to evaluate the multi-key security of the AE algorithms Minalpher and OPP, whose state sizes are both two times larger than the corresponding key sizes. Taking $n = 256$ as an example, we are able to recover almost all masks of a group of 2^{86} independent keys by doing 2^{86} unkeyed queries and 2^{86} queries per key, with 2^{172} (resp. 2^{86}) memory cost in the known-plaintext (resp. blockwise-adaptive chosen-plaintext) model[1]. Accompanied by some brief discussions, we finally conclude the paper in Section 6.

2 Overview of Collision-Based Attacks on Even-Mansour

The new attacks elaborated in this paper combine several published techniques related to collision-based attacks on Even-Mansour construction, which will be described in this section.

2.1 The Basic Collision-Based Attack on Even-Mansour

As a minimalistic design of a block cipher, the Even-Mansour construction is built using an (arbitrary) public permutation P with two whitening keys k_1 and k_2, which is usually defined as:

$$EM_{k_1,k_2}(m) = P(m \oplus k_1) \oplus k_2.$$

There is also a minimal version of using $k = k_1 = k_2$ presented by Dunkelman et al. [DKS12] (denoted by EM for simplicity), which has been proved to be secure up to the same bound as the two-key version.

In [FJM14], Fouque et al. describe a simple collision-based attack, whose idea is to apply the Davies-Meyer construction to EM and to P. Write two functions:

$$F_{EM}(m) = m \oplus EM(m),\ F_P(m) = m \oplus P(m).$$

Note that any collision $F_{EM}(m) = F_P(m')$ indicates that $m \oplus m'$ is a likely candidate for the secret k, and thereby the problem of attacking EM is reduced to the problem of finding a collision between F_{EM} and F_P. As a result, after computing F_{EM} (resp. F_P) on D (resp. T) distinct random values, where $DT \approx 2^{|k|}$, one expects to find the required collisions.

Moreover, as introduced in Section 2.2, the above process can be done in a memory-efficient way by using the distinguished point technique (attributed to R. Rivest and later analyzed in [BPJ99, SRQL02]). First, choose large numbers of random start points from the solution space, and then iterate a pre-defined function on each one of them. As an

[1] For an ideally-secure AE scheme with $n = 2|k|$, adversaries need about $O(2^{n/3} \cdot 2^{n/2}) = O(2^{5n/6})$ time to reveal all of the $O(2^{n/3})$ independent keys. So even if in the known-plaintext attack, our analysis (requiring about $O(2^{2n/3})$ memory) is still effective.

improvement of Hellman's attack [Hel80], this new approach stops each chain once it reaches a set of distinguished points, which are defined according to an easily verifiable condition. For example, if the average length of each chain is $\tilde{l}$, we can choose the set of distinguished points to contain points whose $\log(\tilde{l})$ LSBs are zero.

The distinguished point technique allows to find collisions in a very efficient way. Let g be a function on a set Λ and Λ_0 be a subset of Λ with distinguished points. Starting from a random point $x_0 \in \Lambda$, we build chains by iteratively calculating:

$$x_s = g(x_{s-1}).$$

Once a distinguished point is detected, i.e. $x_l \in \Lambda_0$, we stop constructing the chain by storing only the (x_0, x_l, l) pair and sort the table according to x_l. Two chains ending at the same distinguished point necessarily merge at some point unless one chain is a subchain of the other. The real collision can easily be recovered as soon as a collision in Λ_0 is detected. Indeed, for two colliding chains with length difference l', we rebuild the chain by taking exactly l' steps starting from the longer chain. Based on this point, it suffices to build both chains in parallel until a collision is reached.

Notice that the length of chains is variable, nevertheless, this does not result in a significant penalty on the theoretical time complexity. And conversely the distinguished point method has a big advantage in practical attacks, since we only need to access a pre-computed table once for (about) $\log(\tilde{l})$ evaluations of the iterated function in the online phase. Small number of memory lookups means the feasibility of storing them on hard disk, which is much cheaper than RAM.

2.2 Time/Memory/Data Tradeoff Attacks on Even-Mansour Scheme in the Classical Setting

To attack EM by use of the distinguished point method, Fouque et al. [FJM14] construct a set of chains using the public permutation P and then search for a collision with a chain obtained from the keyed permutation EM. Since different iterated functions result in non-merging chains, it appears that the expected collision cannot be detected efficiently. Here they exploit a subtle property to solve this problem. More specifically, two iterated functions[2] are defined as:

$$\Phi_s = \Phi_{s-1} \oplus EM\left(\Phi_{s-1}\right) \oplus EM\left(\Phi_{s-1} \oplus \delta\right),$$

$$\phi_s = \phi_{s-1} \oplus P\left(\phi_{s-1}\right) \oplus P\left(\phi_{s-1} \oplus \delta\right),$$

where δ is a random non-zero constant and Φ_s (resp. ϕ_s) represents the s-th point on the on-line (resp. off-line) chain. For two points Φ_i and ϕ_j where $\phi_j = \Phi_i \oplus k$, it is not difficult to check that $\phi_{j+1} = \Phi_{i+1} \oplus k$, which implies the same relation remains with the subsequent points, i.e. two chains become parallel.

In order to complete the collision-based attack, we define a distinguished point Φ_i as a point with a value of $EM\left(\Phi_i\right) \oplus EM\left(\Phi_i \oplus \delta\right)$, and define a distinguished point ϕ_j as a point with a value of $P\left(\phi_j\right) \oplus P\left(\phi_j \oplus \delta\right)$. During the preprocessing phase, each off-line chain is evaluated by using iterated function ϕ. Once a distinguished point is detected, we stop constructing this chain by storing $(P\left(\phi_j\right) \oplus P\left(\phi_j \oplus \delta\right), \phi_j)$ pair and sort the table according to the first element. Correspondingly, an on-line chain is created by using iterated function Φ. As long as $EM\left(\Phi_i\right) \oplus EM\left(\Phi_i \oplus \delta\right)$ is matched with an off-line distinguished point $P\left(\phi_j\right) \oplus P\left(\phi_j \oplus \delta\right)$, $\Phi_i \oplus \phi_j$ will be regarded as a candidate value of k.

[2]Despite the fact that these functions are originally used for attacking the two-key Even-Mansour, we point out here they are still applicable for the single-key version. Just as mentioned in [Din15], although simpler definitions, $H_s = EM\left(H_{s-1}\right)$ and $h_s = P\left(h_{s-1}\right)$, can also provide parallelism property, they are permutations (rather than non-bijective mappings) and their behavior cannot be modeled using random functions.

2.3 The Best Previous Attack on Even-Mansour Scheme in the Multi-user Setting

We now summarize the best previous attack by Fouque et al. [FJM14] on Even-Mansour scheme in the multi-user setting, which uses the giant component theorem [Bol01] in graph theory. Their main idea is to construct a graph whose vertices are users and whose edges are labelled with XOR of the corresponding keys. Through both theoretical analysis and experimental verification, they elaborate on the effectiveness and correctness of this attack.

For the sake of simplicity, we once again take the single-key Even-Mansour as an example. When considering the multi-user setting, we suppose that L different users are all using this scheme based on the same permutation P, with each user having its own key $k^{(i)}$, chosen uniformly at random and independently from the others. The off-line chains are created using the same ϕ defined in Section 2.2, while for on-line chains of user $U^{(i)}$, we use the iterated funcion:

$$\Phi_s^{(i)} = \Phi_{s-1}^{(i)} \oplus EM^{(i)}\left(\Phi_{s-1}^{(i)}\right) \oplus EM^{(i)}\left(\Phi_{s-1}^{(i)} \oplus \delta\right),$$

where $EM^{(i)}$ denotes a single-key Even-Mansour with secret key $k^{(i)}$. Similar to the previous analysis, it is easy to examine that two chains built by using $\Phi^{(i)}$ and $\Phi^{(j)}$ can also be parallel and their constant difference is expected to equal $k^{(i)} \oplus k^{(j)}$.

To present a much more efficient tradeoff attack on Even-Mansour by using the graph algorithmic idea, we first construct a graph whose vertices are the users. Afterwards, we compute a small number of on-line chains for every user and a set of off-line chains for the public user (the user with the unkeyed function). For two points $\Phi_u^{(i)}$ and $\Phi_v^{(j)}$, whenever a collision, i.e. $EM^{(i)}\left(\Phi_u^{(i)}\right) \oplus EM^{(i)}\left(\Phi_u^{(i)} \oplus \delta\right) = EM^{(j)}\left(\Phi_v^{(j)}\right) \oplus EM^{(j)}\left(\Phi_v^{(j)} \oplus \delta\right)$, is detected using the distinguished point technique, we add an edge between vertex $U^{(i)}$ and $U^{(j)}$ labelled with $\Phi_u^{(i)} \oplus \Phi_v^{(j)}$ (which is expected to equal $k^{(i)} \oplus k^{(j)}$).

With the increase of the number of edges, a giant component appears in the graph with high probability. Specifically, as shown in [FJM14], we construct a random graph according to the Erdös-Rényi model, in which each possible edge connecting pairs of a given set of L vertices is present, independent of the other edges. In this case, provided that the number of edges $cL/2$ is larger than the number of vertices L, there is with overwhelming probability a single giant component whose size is $(1 - t(c))L$, (see [Bol01]) where

$$t(c) = \frac{1}{c} \sum_{k=1}^{\infty} \frac{k^{k-1}(ce^{-c})^k}{k!}.$$

For instance, once $3L/2$ random edges are generated among the L vertices, it is very likely that 94% of these points are in a large component. To reveal all keys of the users in this set, it suffices to find a collision between the public user and an arbitrary user in the component.

3 Known-Plaintext Attack against TEM-1

In this section, we elaborate on new known-plaintext attacks (considered as the most practical analysis model to a great extent) against TEM-1 in the multi-key setting, where TEM-1 is used under L secret keys, with each key chosen uniformly at random and independently from the others.

3.1 The Details of Our Attack

Our main idea is searching for the linear relation between an arbitrary pair of keys, taking advantage of the giant component theorem in the random graph theory. Specifically, in

a set of L independent keys, we assume the number of message blocks under each key is D. For any $k^{(i)}$, $1 \leq i \leq L$, the encryption result of the s-th message block $m_s^{(i)}$ can be characterized as:

$$c_s^{(i)} \triangleq TEM(k^{(i)}, s, m_s^{(i)}) = P(m_s^{(i)} \oplus f(k^{(i)}, s)) \oplus f(k^{(i)}, s),$$

where $1 \leq s \leq D$, and $f(k^{(i)}, s)$ represents the whitening keys at the s-th step. After applying the Davies-Meyer construction to TEM-1, it holds that:

$$m_s^{(i)} \oplus c_s^{(i)} = P(m_s^{(i)} \oplus f(k^{(i)}, s)) \oplus m_s^{(i)} \oplus f(k^{(i)}, s).$$

Any collision between $m_u^{(i)} \oplus c_u^{(i)}$ and $m_v^{(j)} \oplus c_v^{(j)}$ indicates that $m_u^{(i)} \oplus f(k^{(i)}, u)$ is a likely candidate value of $m_v^{(j)} \oplus f(k^{(j)}, v)$. Likewise, any collision between $m_u^{(i)} \oplus c_u^{(i)}$ and $P(x_v) \oplus x_v$ means that $m_u^{(i)} \oplus x_v$ is expected to equal $f(k^{(i)}, u)$.

With this idea in mind, the problem of revealing all of the keys is reduced to searching for enough linear relations among them and then solving the system by finding a collision (or rather a few collisions) between off-line evaluations and on-line queries. For this, we utilize the graph algorithmic idea introduced in Section 2.3 to construct the system of linear equations. The procedure of our attack is shown below.

(1) For L independent keys, store $(m_s^{(i)}, s)$, $1 \leq i \leq L$, $1 \leq s \leq D$, in an ordered table which is sorted according to the value of $m_s^{(i)} \oplus c_s^{(i)}$.

(2) Create a graph whose vertices represent all of the keys. Search for collisions between any two keys. As soon as $m_u^{(i)} \oplus c_u^{(i)} = m_v^{(j)} \oplus c_v^{(j)}$ is found, we add an edge between $k^{(i)}$ and $k^{(j)}$ labelled with $m_u^{(i)} \oplus m_v^{(j)}$ which is likely equal to $f(k^{(i)}, u) \oplus f(k^{(j)}, v)$.

(3) During the off-line phase, randomly choose T inputs x_s, $1 \leq s \leq T$, and calculate $y_s = P(x_s)$. After each evaluation, search for $x_s \oplus y_s$ from the table constructed in step (1). Once we detect $x_u \oplus y_u = m_v^{(j)} \oplus c_v^{(j)}$, $x_u \oplus m_v^{(j)}$ is probably a candidate value of $f(k^{(j)}, v)$.

(4) Test $f(k^{(j)}, v)$ obtained in step (3) using another existing $(m_{v'}^{(j)}, c_{v'}^{(j)})$ where $v' \neq v$. If $c_{v'}^{(j)} = P(m_{v'}^{(j)} \oplus f(k^{(j)}, v')) \oplus f(k^{(j)}, v')$, go to step (5). Otherwise return step (3) to search for another collision.

(5) Starting from the verified $f(k^{(j)}, v)$, we solve the system of linear equations which is generated in step (2). Note that for the new candidate value obtained at each step, we need to use a trial encryption (similar to the approach in step (4)) to insure its correctness.

3.2 The Analysis of Our Attack

Given L, D and T, it is necessary to estimate the number of collisions which can be found in step (2), since this value not only dominates the time complexity of solving the connected system, but also determines the number of keys which can be recovered. Indeed, for two random pairs, $(m_u^{(i)}, u)$ and $(m_v^{(j)}, v)$, the probability that $m_u^{(i)} \oplus f(k^{(i)}, u)$ equals $m_v^{(j)} \oplus f(k^{(j)}, v)$ (so that $m_u^{(i)} \oplus c_u^{(i)} = m_v^{(j)} \oplus c_v^{(j)}$) is $1/2^n$. In addition, by a reasonable randomness assumption[3], the probability of $m_u^{(i)} \oplus c_u^{(i)} = m_v^{(j)} \oplus c_v^{(j)}$ conditioned on $m_u^{(i)} \oplus f(k^{(i)}, u) \neq m_v^{(j)} \oplus f(k^{(j)}, v)$ is also $1/2^n$. Consequently, the expected number of collisions in step (2) is:

[3] A similar randomness assumption is also applied to the analysis of LEX in [DK13]. As pointed out by Dunkelman et al., such an assumptions is very delicate, so it is very important to check its validity in each concrete case of study.

$$Num[Coll] = \begin{pmatrix} L \\ 2 \end{pmatrix} \times D^2 \times \left[\tfrac{1}{2^n} + \left(1 - \tfrac{1}{2^n}\right) \times \tfrac{1}{2^n}\right].$$

The number of desirable collisions (i.e. $m_u^{(i)} \oplus c_u^{(i)} = m_v^{(j)} \oplus c_v^{(j)}$ due to $m_u^{(i)} \oplus f(k^{(i)}, u) = m_v^{(j)} \oplus f(k^{(j)}, v)$) is $L(L-1)D^2/2^{n+1}$, which accounts for approximately half of $Num[Coll]$. In other words, a little more than half of the linear relations we construct in step (2) are correct, which implies we will abandon nearly half linear relations by use of trial encryptions. Thereby, based on the giant connected component theorem, as long as we select parameters such that

$$\frac{L(L-1)D^2}{2^{n+1}} \geq cL,$$

where c is a small constant number, then almost all keys are in the component with correct edges. With the help of collisions found in step (3), we are finally able to reveal all keys in the connected system.

Despite being more practical, the known-plaintext attack inevitably requires plenty of memory. For instance, with $O(2^{n/3})$ queries per key and $O(2^{n/3})$ off-line calculations, the overall memory complexity of our attack to recover all of $O(2^{n/3})$ keys is $O(2^{2n/3})$.

3.3 Experimental Results

To confirm the validity of our known-plaintext attack on TEM-1, we implemented it on a standard PC with the lightweight block cipher SIMON32 [BSS+13] as the public permutation. Notice that our algorithm above is applicable for any linear function $f(k, s)$ of k. Here we particularly selected $f(k, s) = \alpha^s k + s$ where $s = 1, 2 \cdots$, and α is primitive element over $GF(2^{32})$ with minimal polynomial $x^{32} + x^7 + x^6 + x^2 + 1$. We simulated 2^{11} independent keys and for each key we queried a random message string of length 3,550 ($\approx \sqrt{3} \times 2^{11}$). Experimentally we found in total 12,336 collisions in step (2). After throwing away 6,122 wrong linear relations by using another trial encryption, the remaining 6,214 correct edges constituted a giant component containing 2,043 keys. Eventually, with $2,077$ off-line evaluations we deduced all keys in the system ($2,043 \approx 2,048 \times 99.7\%$), which is basically consistent with the theoretical analysis with $c = 3$.

4 Adaptive Chosen-Plaintext Attack on TEM-1

In this section we restrict the linear function f to $f(k, s) = \beta\alpha^s k$, where $s = 1, 2, \cdots$, and α, β are two arbitrary invertible linear transformations. Such f has been widely used in the design of tweakable Even-Mansour schemes, for its convincing mathematical property in the security analysis and convenient implementations. The most prominent example may be the MEM construction [GJMN16] proposed at EUROCRYPT 2016. Also, this tweak form is popular in the design of tweakable blockciphers (rather than TEM), such as OCB2 [Rog04] and some CAESAR candidates [CAE14], e.g. COPA, ELmD, OTR, POET and SHELL.

We describe multi-key attacks on this specific TEM-1 in the adaptive chosen-plaintext setting, which is much more generous with adversaries than the known-plaintext model. Compared with the attack of Section 3, we can drastically reduce the memory cost with other parameters unchanged.

4.1 New Technique in Our Attack

Based on the expression of TEM-1 and the restriction on the linear fucntion, the encryption result of the s-th message block m_s can be denoted by:

$$TEM(k, s, m_s) = P(m_s \oplus \beta\alpha^s k) \oplus \beta\alpha^s k. \tag{1}$$

In order to attack this scheme using distinguished point technique, we need to define appropriate iterated functions such that a collision, whose specific form will be given later, between on-line points and off-line ones can be efficiently detected. However, unlike the case described in Section 2.2, two chains will never have a constant difference between them due to the existence of tweaks. In order to solve this dilemma, we first randomly select a non-zero constant δ and then define the on-line function as:

$$\Theta_s = \Theta_{s-1} \oplus \alpha^{-s} \cdot TEM\left(k, s, \beta\alpha^s\Theta_{s-1}\right) \oplus \alpha^{-s} \cdot TEM\left(k, s, \beta\alpha^s\Theta_{s-1} \oplus \delta\right), \quad (2)$$

where the startpoint Θ_0 is randomly selected. More precisely, while calculating the s-th point Θ_s from the $(s-1)$-th point Θ_{s-1} on a Θ chain, we have to query message blocks $\beta\alpha^s\Theta_{s-1}$ and $\beta\alpha^s\Theta_{s-1} \oplus \delta$ with the same tweak (i.e. whitening keys used are both $\beta\alpha^s k$). For a better understanding, we depict the process of creating Θ_1 and Θ_s on a Θ chain in Figure 2.

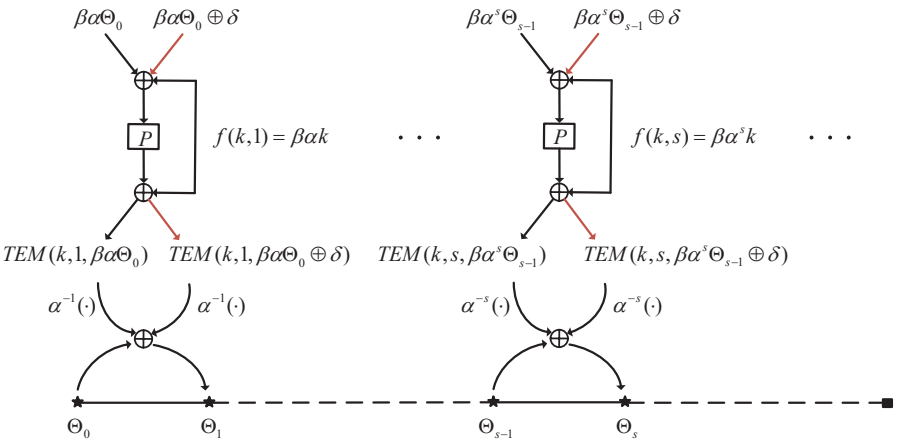

Figure 2: The Procedure of constructing two points, Θ_1 and Θ_s, on an on-line chain Θ

Similarly, for the off-line evaluation, we make use of the iterated function defined as follows:

$$\theta_s = \theta_{s-1} \oplus \alpha^{-s} \cdot P\left(\beta\alpha^s\theta_{s-1}\right) \oplus \alpha^{-s} \cdot P\left(\beta\alpha^s\theta_{s-1} \oplus \delta\right), \quad (3)$$

where θ_0 is the startpoint chosen at random. We remark that our construction above suffices to get candidate values for the secret k. Indeed, let two points Θ_{u-1} and θ_{v-1} belong to a Θ chain and a θ chain respectively. Assume

$$\alpha^u\Theta_{u-1} \oplus \alpha^v\theta_{v-1} = \alpha^u k, \quad (4)$$

and using (1), it holds that:

$$\begin{aligned} & TEM(k, u, \beta\alpha^u\Theta_{u-1}) \oplus TEM(k, u, \beta\alpha^u\Theta_{u-1} \oplus \delta) \\ &= P\left(\beta\alpha^u\Theta_{u-1} \oplus \beta\alpha^u k\right) \oplus P\left(\beta\alpha^u\Theta_{u-1} \oplus \beta\alpha^u k \oplus \delta\right) \\ &= P\left(\beta\alpha^v\theta_{v-1}\right) \oplus P\left(\beta\alpha^v\theta_{v-1} \oplus \delta\right). \end{aligned} \quad (5)$$

Apply $\beta\alpha\beta^{-1}$ on both sides of (4): $\beta\alpha\beta^{-1}(\beta\alpha^u\Theta_{u-1} \oplus \beta\alpha^v\theta_{v-1}) = \beta\alpha\beta^{-1}(\beta\alpha^u k)$, and then obtain

$$\alpha^{u+1}\Theta_{u-1} \oplus \alpha^{v+1}\theta_{v-1} = \alpha^{u+1}k. \quad (6)$$

According to (2) and (3), the next elements, Θ_u and θ_v, in each of these chains will satisfy:

$$\alpha^{u+1}\Theta_u = \alpha^{u+1}\Theta_{u-1} \oplus \alpha \cdot TEM\left(k, u, \beta\alpha^u\Theta_{u-1}\right) \oplus \alpha \cdot TEM\left(k, u, \beta\alpha^u\Theta_{u-1} \oplus \delta\right),$$

$$\alpha^{v+1}\theta_v = \alpha^{v+1}\theta_{v-1} \oplus \alpha \cdot P\left(\beta\alpha^v\theta_{v-1}\right) \oplus \alpha \cdot P\left(\beta\alpha^v\theta_{v-1} \oplus \delta\right).$$

Furthermore, by use of (5) and (6), we have

$$\alpha^{u+1}\Theta_u \oplus \alpha^{v+1}\theta_v = \alpha^{u+1}\Theta_{u-1} \oplus \alpha^{v+1}\theta_{v-1} = \alpha^{u+1}k,$$

which implies

$$TEM(k, u{+}1, \beta\alpha^{u+1}\Theta_u) \oplus TEM(k, u{+}1, \beta\alpha^{u+1}\Theta_u \oplus \delta) = P\left(\beta\alpha^{v+1}\theta_v\right) \oplus P\left(\beta\alpha^{v+1}\theta_v \oplus \delta\right).$$

As a consequence, as soon as by chance $\beta\alpha^u\Theta_{u-1} \oplus \beta\alpha^v\theta_{v-1} = \beta\alpha^u k$ where Θ_{u-1} is an element of a Θ chain and θ_v is an element of a θ chain, the subsequent points of these chains will keep a favorable relation $\beta\alpha^{u+\tau}\Theta_{u+\tau-1} \oplus \beta\alpha^{v+\tau}\theta_{v+\tau-1} = \beta\alpha^{u+\tau}k$ where $\tau = 1, 2, 3, \cdots$.

The detection of the above phenomenon is compatible with the distinguished point technique. For the on-line chain it suffices to define a distinguished point Θ_{u-1} as a point with a value of $TEM(k, u, \beta\alpha^u\Theta_{u-1}) \oplus TEM(k, u, \beta\alpha^u\Theta_{u-1} \oplus \delta)$. With respect to chains built off-line, we define a distinguished point θ_{v-1} as a point with a value of $P\left(\beta\alpha^v\theta_{v-1}\right) \oplus P\left(\beta\alpha^v\theta_{v-1} \oplus \delta\right)$. Once $\alpha^u\Theta_{u-1} \oplus \alpha^v\theta_{v-1} = \alpha^u k$ and $\theta_{v-1+\tau}$ is detected as a distinguished point in a θ chain, $\Theta_{u-1+\tau}$ is also a distinguished point in the Θ chain (i.e. these two distinguished points collide), and thereby $\alpha^{-(u+\tau)}(\alpha^{u+\tau}\Theta_{u-1+\tau} \oplus \alpha^{v+\tau}\theta_{v-1+\tau})$ provides a candidate value for k.

A significant difference compared with the methodology of Fouque et al. is that chains created by our iterated functions become no longer parallel. Nevertheless, it has no influence on the key-recovery attack. Through skillfully choosing queries of each step (rather than simply inquiring the preceding point), we guarantee that the desired collision can be efficiently detected with appropriate definitions of distinguished points, which suffices to suggest the secret key.

4.2 Experimental Results

We implemented our attack on the specific TEM-1 to confirm its validity. Again, SIMON32 with a fixed key was used as the internal permutation, but unlike the experiment in Section 3.2, hereon we chose $f(k, s) = \beta\alpha^s k$, where $s = 1, 2, \cdots$, and β, α are two elements over $GF(2^{32})$ (where the associated irreducible polynomial is $x^{32} + x^7 + x^6 + x^2 + 1$). Through using distinguished points containing 13 zeroes and bounding the length of chains to 2^{15}, in all we generated 71 available off-line chains after throwing away 48 merging chains and abandoning one chain suspected to contain a loop. On the other side, the process of constructing one on-line chain was carried out 100,000 times, where we could find a collision between off-line distinguished points and on-line distinguished point 53,605 times, among which we successfully recovered the correct key 53,604 times. This result basically coincides with the birthday paradox as the number of points evaluated off-line is approximately 2^{13+6} and the on-line chain is of average length 2^{13}.

4.3 The Procedure of Our Attack

We now present an adaptive chosen-plaintext attack on the specific TEM-1 in the multi-key setting, taking advantage of the technique shown in Section 4.1 and the methodology of Fouque et al. presented in Section 2.3. The main idea is to construct chains by using a function based on this TEM-1 and then search for collisions between them. In a set of L independent keys, we define the iterated function for $k^{(i)}$ as:

$$\Theta_s^{(i)} = \Theta_{s-1}^{(i)} \oplus \alpha^{-s} \cdot TEM\left(k^{(i)}, s, \beta\alpha^s\Theta_{s-1}^{(i)}\right) \oplus \alpha^{-s} \cdot TEM\left(k^{(i)}, s, \beta\alpha^s\Theta_{s-1}^{(i)} \oplus \delta\right),$$

where $1 \le i \le L$ and $s = 1, 2, 3, \cdots$. Correspondingly, to create off-line chains we define the unkeyed iterated function for $k^{(0)}$ as:

$$\theta_s = \theta_{s-1} \oplus \alpha^{-s} \cdot P\left(\beta\alpha^s\theta_{s-1}\right) \oplus \alpha^{-s} \cdot P\left(\beta\alpha^s\theta_{s-1} \oplus \delta\right).$$

For on-line chains of $k^{(i)}$, we define a distinguished point $\Theta_{u-1}^{(i)}$ as a point with a value of $TEM(k^{(i)}, u, \beta\alpha^u\Theta_{u-1}^{(i)}) \oplus TEM(k^{(i)}, u, \beta\alpha^u\Theta_{u-1}^{(i)} \oplus \delta)$, while for off-line chains, we define a distinguished point θ_{v-1} as a point with a value of $P\left(\beta\alpha^v\theta_{v-1}\right) \oplus P\left(\beta\alpha^v\theta_{v-1} \oplus \delta\right)$. On the one hand, whenever a collision between on-line distinguished points among different keys, $\Theta_{u'}^{(i)}$ and $\Theta_{v'}^{(j)}$, is detected, $\alpha^{u'+1}\Theta_{u'}^{(i)} \oplus \alpha^{v'+1}\Theta_{v'}^{(j)}$ provides a candidate value for $\alpha^{u'+1}k^{(i)} \oplus \alpha^{v'+1}k^{(j)}$. On the other hand, once an on-line distinguished point $\Theta_{u'}^{(i)}$ is matched with an off-line distinguished point $\theta_{v'}$, $\alpha^{u'+1}\Theta_{u'}^{(i)} \oplus \alpha^{v'+1}\theta_{v'}$ is expected to equal $\alpha^{u'+1}k^{(i)}$. Therefore our attack works as follows:

(1) Construct a graph whose vertices are labelled by the L independent keys and $k^{(0)}$.

(2) For each $k^{(i)}$, where $1 \leq i \leq L$, start from a random startpoint $\Theta_0^{(i)}$ and then create a constant number $c/2$ of chains using $\Theta^{(i)}$.

(3) Build several chains for $k^{(0)}$ starting from an arbitrary startpoint through iteratively computing θ_s.

(4) Search for collisions between on-line distinguished points among all keys. As soon as a collision between $\Theta_{u'}^{(i)}$ and $\Theta_{v'}^{(j)}$ is found, we add an edge between $k^{(i)}$ and $k^{(j)}$ labelled with $\alpha^{u'+1}\Theta_{u'}^{(i)} \oplus \alpha^{v'+1}\Theta_{v'}^{(j)}$, which is expected to equal $\alpha^{u'+1}k^{(i)} \oplus \alpha^{v'+1}k^{(j)}$.

(5) Search for collisions of distinguished points between all keys and $k^{(0)}$. Whenever a collision between $\Theta_{u'}^{(i)}$ and $\theta_{v'}$ is detected, we add an edge between $k^{(i)}$ and $k^{(0)}$ labelled with $\alpha^{u'+1}\Theta_{u'}^{(i)} \oplus \alpha^{v'+1}\theta_{v'}$, which will be considered to equal $\alpha^{u'+1}k^{(i)}$.

(6) With only a single collision of distinguished points between $k^{(0)}$ and any of the keys in the giant component, we are able to recover all keys in this connected system.

According to the typical parameters in [FJM14], we expect with $2^{n/3}$ independent keys in total, $c \cdot 2^{n/3}$ queries per key (where c is a small arbitrary constant) and $2^{n/3}$ unkeyed queries, to recover almost all the $2^{n/3}$ keys with overwhelming probability. For instance, approximately 98% of the $2^{n/3}$ keys can be revealed while $c = 4$.

5 Mask-Recovery Attacks on Minalpher and OPP in the Multi-key Setting

In this section, we apply our techniques introduced in Section 3 and Section 4 to analyze authenticated encryption algorithms Minalpher and OPP. All our attacks are mounted in the multi-user setting, where Minalpher (resp. OPP) is used under L keys, with each key chosen uniformly at random and independently from the others.

5.1 Authenticated Encryption Algorithm Minalpher and OPP

Minalpher is a second-round candidate of CAESAR competition proposed by Sasaki et al. in an attempt to provide 128-bit security for both of confidentiality and integrity. Not only does it provide some level of the robustness against nonce misuse and unverified plaintext release, it also enjoys an attractive advantage on various platforms, especially in embedded systems. Minalpher supports two modes of operation: message authentication code (MAC) and authenticated encryption with associated data (AEAD). Different functionalities as

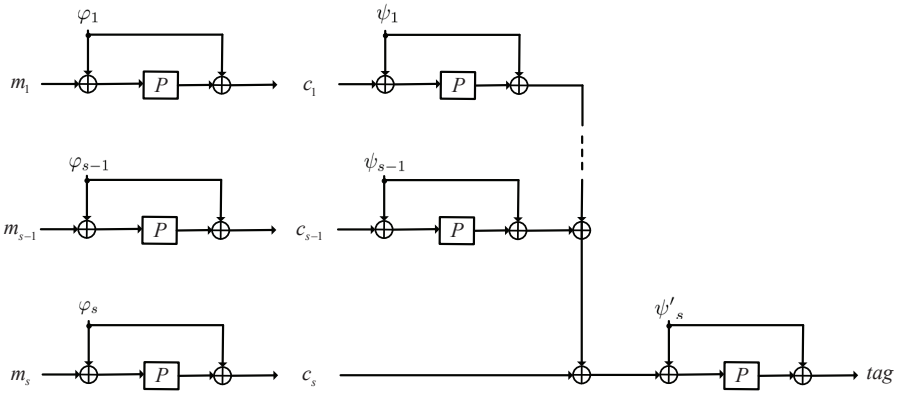

Figure 3: AEAD Mode of Minalpher with Empty Auxiliary Data and Non-empty Message

these modes provide, they have very similar designs, and our focus is mainly placed on the AEAD mode in the subsequent discussions.

Let $k \in \{0,1\}^{n/2}$ be a secret key, $N \in \{0,1\}^{n/2-s}$ be a nonce and $flag \in \{0,1\}^s$. In Minalpher, the recommended value of parameters are $n = 256$ and $s = 24$. Namely, the sizes of the internal permutation and a nonce are 256 bits and 104 bits respectively.

Before the authenticated encryption procedure, internal state is initialized with the string $k||flag||N$, which is then updated by applying the (involution) public permutation Minalpher-P.[4] The corresponding result, usually referred to as a mask and here denoted by Q, is used for the later computations:

$$Q = (k||flag||N) \oplus P(k||flag||N).$$

The procedure of authenticated encryption is visualized in Figure 3, where $\varphi_i = y^{2i-1}Q$, $\psi_i = y^{2i}Q$ and $\psi'_s = y^{2s-1}(y+1)Q$. Here y is a primitive element of $GF(2^{256})$,[5] and we ignore the process related to any auxiliary data since it is irrelevant to our attack.

Based on the description above, we know the fundamental component of Minalpher is a tweakable Even-Mansour primitive, which is used for processing message blocks in parallel. Meanwhile, it is not difficult to see that TEM-1 we study in this present paper naturally generalizes the tweakable Even-Mansour scheme adopted in Minalpher. Specifically, we have

$$TEM(k, s, m_s) = P(m_s \oplus f(Q, s)) \oplus f(Q, s),$$

where $s = 1, 2, \cdots$, $f(Q, s) = y^{2s-1}Q$, and $Q = (k||flag||N) \oplus P(k||flag||N)$.

Offset Public Permutation (OPP) mode is a nonce-respecting AE scheme proposed by Granger et al. at EUROCRYPT 2016. One of the most appealing features of OPP is the generation of tweak-dependent mask, which combines the best of both word-oriented LFSR-based and powering-up-based masking approaches. As a generalization of OCB3 [KR11] to arbitrary block sizes, it obtains remarkable improvements in the simplicity and efficiency. Instantiated with a reduced-round BLAKE2b permutation [ANWW13], OPP achieves a peak speed on an Intel Haswell processor, which is faster than any other permutation-based CAESAR submission. The designers claim that OPP behaves like a random AE up to attack complexity dominated by $\min\{2^{n/2}, 2^{|k|}\}$, where n is the size of the permutation and $|k|$ is the key length.

[4]Since our attack is structural, it is independent of the particular choices of the public permutation P of Minalpher. Thus, we refer the interested reader to [STA+15] for specific description.

[5]More precisely, designers of Minalpher represent $GF(2^{256})$ with a tower of extensions using two irreducible polynomials, whose specific forms can be found in [STA+15].

Let k and N be a secret key and a nonce respectively. Similar to the brief introduction of Minalpher, we here only describe the encryption process of OPP without auxiliary data in Figure 4, where $Q = P(k||N)$, and φ, $\varphi_2 = \varphi^2 + \varphi + Id$ are essentially two invertible linear transformations, whose concrete expressions are specified in [GJMN16]. Again, we recognize the tweakable Even-Mansour construction employed in OPP is a specific case of our TEM-1 scheme. More precisely, we have

$$TEM(k, s, m_s) = P(m_s \oplus f(Q, s)) \oplus f(Q, s),$$

where $s = 1, 2 \cdots$ and $f(Q, s) = \varphi_2 \varphi^{s-1} Q$.

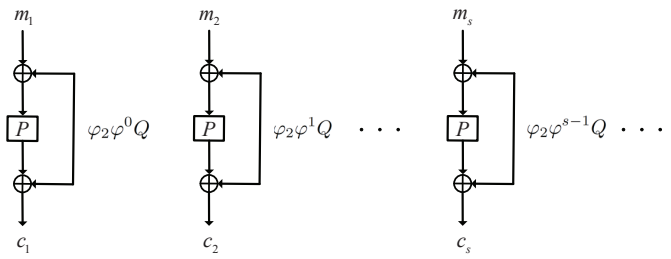

Figure 4: Encryption Procedure of OPP with Empty Associated Data

5.2 Known-Plaintext Attacks against Minalpher and OPP

Our multi-key analysis on TEM-1 in the known-plaintext setting is applicable for any function $f(k, t)$ linear in k. Therefore, we can directly utilize the attack algorithm (without any modification) to evaluate the multi-key security of Minalpher and OPP, where our recovery goal is the mask under each independent key.

Now taking Minalpher as an example, we are able to recover almost all masks (i.e. Q's) of a group of 2^{86} independent keys by doing 2^{86} unkeyed queries and 2^{86} queries per key. There certainly exist many other tradeoffs between the number of independent keys and the on-line/off-line queries. For instance, provided that the goal is to reveal almost all masks of 2^{64} independent keys, we need approximately 2^{96} queries per key and 2^{96} unkeyed queries.

It is necessary for us to point out the significance of mask-recovery attack against AE scheme, despite the fact that we do not continue to tentatively reveal the secret key. Indeed, for any $k^{(i)}$, once we obtain $Q^{(i)}$ which is derived from a certain $(k^{(i)}, N^{(i)})$, we are able to achieve the associated ciphertext of arbitrary message string without even inquiring the encryption oracle. Furthermore, we can make valid forgeries of any form under this key/nonce pair. As a result, mask-recovery is to a great extent equivalent to revealing the secret key for large numbers of practical application scenarios.

5.3 Blockwise-Adaptive Chosen-Plaintext Attacks on Minalpher and OPP

We now elaborate on mask-recovery attacks on AE algorithms Minalpher and OPP in the blockwise-adaptive chosen-plaintext setting [Bar06, JMV02, FJP04], taking advantage of the technique shown in Section 4.1 and the methodology of Fouque et al. described in Section 2.3. The main idea is to construct chains by using a function based on Minalpher (resp. OPP) and then search for collisions between them.

For Minalpher, the encryption result of the s-th message block m_s can be expressed as follows:

$$TEM(Q, s, m_s) = P(m_s \oplus f(Q, s)) \oplus f(Q, s),$$

where $s = 1, 2, \cdots$ and $f(Q, s) = y^{2s-1}Q$. To apply our attack of Section 4.3, we just need to replace β and α by y^{-1} and y^2 respectively. As a downside, this attack requires to reuse nonce to ensure that the mask (i.e. (k, N) pair) under each independent key is unchanged even if we construct only one on-line chain for this key. Specifically, while calculating the s-th point $\Theta_s^{(i)}$ from the $(s-1)$-th point $\Theta_{s-1}^{(i)}$ on a $\Theta^{(i)}$ chain, we need to query message block $y^{2s-1}\Theta_{s-1}^{(i)}$ and $y^{2s-1}\Theta_{s-1}^{(i)} \oplus \delta$ under the same $f(Q^{(i)}, s) = y^{2s-1}Q^{(i)}$, which is forbidden in the nonce-respecting model.

As a consequence, only Minalpher with nonce reuse can be analyzed by utilizing the technique of Section 4.3. To provide multi-key analysis of Minalpher and OPP in the nonce-respecting setting, we propose new iterated functions for $Q^{(i)}$ as:

$$\Theta_s^{(i)} = \Theta_{s-1}^{(i)} \oplus \alpha^{-s}\beta^{-1} \cdot TEM\left(Q^{(i)}, s, \beta\alpha^s\Theta_{s-1}^{(i)}\right) \oplus \alpha^{-(s+1)}\beta^{-1} \cdot TEM\left(Q^{(i)}, s+1, \beta\alpha^{s+1}\Theta_{s-1}^{(i)}\right)$$

where $TEM(Q^{(i)}, s, \beta\alpha^s\Theta_{s-1}^{(i)}) = P(\beta\alpha^s\Theta_{s-1}^{(i)} \oplus \beta\alpha^s Q^{(i)}) \oplus \beta\alpha^s Q^{(i)}$. Keeping in line with the restriction in Section 4, α and β are also two arbitrary invertible linear transformations.

Likewise, in order to create off-line chains for $k^{(0)}$, we use the function defined as follows:

$$\theta_s = \theta_{s-1} \oplus \alpha^{-s}\beta^{-1} \cdot P(\beta\alpha^s\theta_{s-1}) \oplus \alpha^{-(s+1)}\beta^{-1} \cdot P(\beta\alpha^{s+1}\theta_{s-1}).$$

We remark that the collisions between on-line and off-line distinguished points suffice to get candidate values for the mask. Indeed, as soon as by chance

$$\alpha^u\Theta_{u-1}^{(i)} \oplus \alpha^v\theta_{v-1} = \alpha^u Q^{(i)},$$

where $\Theta_{u-1}^{(i)}$ (resp. θ_{v-1}) belongs to a $\Theta^{(i)}$ (resp. θ) chain, the subsequent points of these two chains will keep a favorable relation:

$$\alpha^{u+\tau}\Theta_{u+\tau-1}^{(i)} \oplus \alpha^{v+\tau}\theta_{v+\tau-1} = \alpha^{u+\tau}Q^{(i)},$$

where $\tau = 1, 2, 3, \cdots$.

In this case, we define an on-line distinguished point $\Theta_{u-1}^{(i)}$ as a point with a value of $\beta^{-1} \cdot TEM\left(Q^{(i)}, u, \beta\alpha^u\Theta_{u-1}^{(i)}\right) \oplus \alpha^{-1}\beta^{-1} \cdot TEM\left(Q^{(i)}, u+1, \beta\alpha^{u+1}\Theta_{u-1}^{(i)}\right)$. Also, an off-line distinguished point θ_{v-1} is defined as a point with a value of $\beta^{-1} \cdot P(\beta\alpha^v\theta_{v-1}) \oplus \alpha^{-1}\beta^{-1} \cdot P(\beta\alpha^{v+1}\theta_{v-1})$. Similar to the analysis in Section 4.1 and Section 4.3, whenever a collision between on-line distinguished points among different users, $\Theta_{u'}^{(i)}$ and $\Theta_{v'}^{(j)}$, is detected, $\alpha^{u'+1}\Theta_{u'}^{(i)} \oplus \alpha^{v'+1}\Theta_{v'}^{(j)}$ is expected to equal $\alpha^{u'+1}Q^{(i)} \oplus \alpha^{v'+1}Q^{(j)}$. As soon as an on-line distinguished point $\Theta_{u'}^{(i)}$ is matched with an off-line distinguished point $\theta_{v'}$, $\alpha^{u'+1}\Theta_{u'}^{(i)} \oplus \alpha^{v'+1}\theta_{v'}$ will provide a candidate value for $\alpha^{u'+1}Q^{(i)}$.

Compared with the iterated functions used in Section 4.3, an obvious benefit of our new definitions is that we are able to build one on-line chain without nonce reuse. Nevertheless to apply the previous attack in the nonce-respecting setting, it is not enough to simply replace the iterated functions. The reason is that we have to create a small constant number of chains (rather than only one chain) for each independent mask to ensure there is a single giant component in the graph with high probability. Hereon, we consider to make full use of the advantage of the blockwise-adaptive chosen-plaintext setting to address this issue. Specifically, for each independent mask, an on-line chain is not terminated when reaching the first distinguished point. By taking a random value as the next point on this chain, we continue the iteration up to the second distinguished point. After repeating this process several times, we can eventually concatenate originally independent chains into one long chain, which makes it possible for us to respect the nonce in the multi-user setting. For the sake of clarity, the process of creating $\Theta_s^{(i)}$ on a $\Theta^{(i)}$ chain with three distinguished points is visualized in Figure 5.

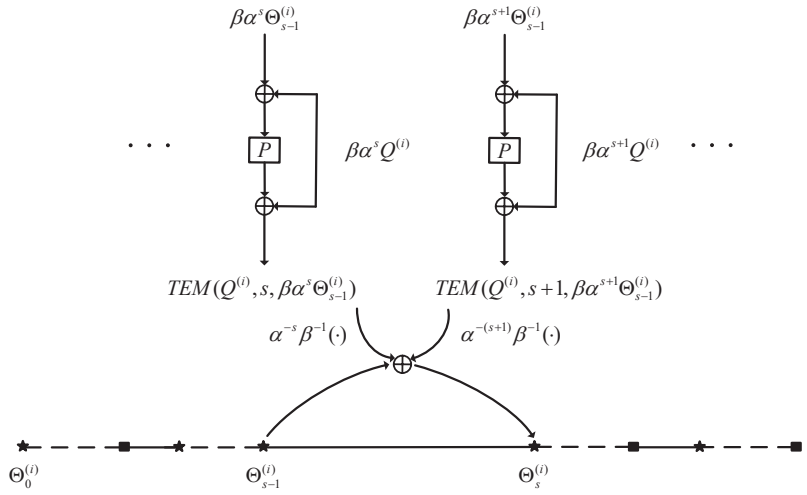

Figure 5: The Process of creating $\Theta^{(i)}_s$ on a $\Theta^{(i)}$ chain with three distinguished points

Remark 1. The nonce-respecting attack above is not applicable to non-blockwise-adaptive adversaries. Nevertheless, if allowed to reuse the nonce, our analysis can be easily adjusted to such adversary. The core idea is to divide a long chain of a single query into short chains of several short queries, whose lengths are quite small. For instance, we first query m_1, m_2 and obtain c_1, c_2. Under the same nonce, then query m'_1, m'_2, where m'_1 is computed based on c_1. After repeating this process $2^{n/3}$ times, we finally get $2^{n/3}$ online chains (each is of length 2), which makes it possible to provide a non-blockwise-adaptive analysis.

5.4 Experimental Results

Taking Minalpher as our target, we implemented the above attack using SIMON32 instead of the 256-bit permutation. We simulated 2^{11} independent masks and for each mask we built one chain with 3 distinguished points. By using distinguished points containing 11 zeroes and bounding the length of chains to 3×2^{13}, we successfully generated available chains for all masks. Namely, we were always able to find at least one distinguished point on the chain of each mask after 3×2^{13} on-line evaluations. Experimentally, the size of the giant connected component contained 1,999 masks. With a single off-line chain of length $2,027$, we eventually deduce all masks in the component, which is basically consistent with the theoretical analysis.

6 Conclusion and Future Work

In the present paper, we introduce multi-key collision-based attacks against TEM-1, in both known-plaintext and adaptive chosen-plaintext setting. Some of these attacks are surprisingly efficient, and despite their limitations, we believe they demonstrate the small security margin of TEM-1-based AE schemes against multi-key attacks. By such analysis, we want to raise an alert that public-permutation-based modes seem to be weaker than blockcipher-based modes in the multi-key setting, where the latter takes advantages that blockciphers with independent keys are usually assumed to be independent pseudorandom permutations.

In the multi-key setting, Mouha et al. [ML15] prove that the Even-Mansour block cipher is secure up to $(\tilde{D}^2L + 2\tilde{D}T)/2^n$, where $\tilde{D}$ (resp. T) represents the total number of keyed (resp. unkeyed) queries. It is not clear whether TEM-1 has the same security bound,

and we are not able to show that our generic attack is optimal. Further exploration of these questions is left to future work.

Acknowledgments. The authors would like to thank all anonymous referees for their valuable comments that greatly improve the manuscript. We are also grateful to Si Gao for providing useful suggestions on the related experiments. This work is supported by the National Basic Research Program of China (No.2013CB338002) and National Natural Science Foundation of China (No.61272476, No.61672509, No.61572484).

References

[ABP+13] N.J. AlFardan, D.J. Bernstein, K.G. Paterson, B. Poettering, and J.C.N. Schuldt. On the Security of RC4 in TLS. In *Proceedings of the 22th USENIX Security Symposium*, pages 305–320, 2013.

[ANWW13] J. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein. BLAKE2: Simpler, Smaller, Fast as MD5. In *ACNS 2013*, volume 7954 of *Lecture Notes in Computer Science*, pages 119–135. Springer, 2013.

[Bar06] G.V. Bard. Modes of Encryption Secure against Blockwise-Adaptive Chosen-Plaintext Attack. *IACR Cryptology ePrint Archive*, 2006:271, 2006.

[BB12] W.C. Barker and E. Barker. Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf. 2012.

[BDJR97] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A Concrete Security Treatment of Symmetric Encryption. In *FOCS 1997*, pages 394–403, 1997.

[BK09] A. Biryukov and D. Khovratovich. Related-Key Cryptanalysis of the Full AES-192 and AES-256. In *ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2009.

[Bol01] B. Bollobás. Random Graphs (2nd Edition). Cambridge Studies in Advanced Mathematics. 2001.

[BPJ99] J. Borst, B. Preneel, and V. Joos. On the Time-Memory Tradeoff Between Exhaustive Key Search and Table Precomputation. *Radiat Prot Dosimetry*, 31(1-4):193–197, 1999.

[BSS+13] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The SIMON and SPECK Families of Lightweight Block Ciphers. *IACR Cryptology ePrint Archive*, 2013:404, 2013.

[BW00] A. Biryukov and D. Wagner. Advanced Slide Attacks. In *EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 589–606. Springer, 2000.

[CAE14] CAESAR. Competition for Authenticated Encryption: Security, Applicability, and Robustness. http://competitions.cr.yp.to/caesar.html. 2014.

[CLS15] B. Cogliati, R. Lampe, and Y. Seurin. Tweaking Even-Mansour Ciphers. In *CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 189–208. Springer, 2015.

[CMS11] S. Chatterjee, A. Menezes, and P. Sarkar. Another Look at Tightness. In *SAC 2011*, volume 7118 of *Lecture Notes in Computer Science*, pages 293–319. Springer, 2011.

[CS15a] B. Cogliati and Y. Seurin. Beyond-Birthday-Bound Security for Tweakable Even-Mansour Ciphers with linear tweak and key mixing. In *ASIACRYPT 2015, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 134–158. Springer, 2015.

[CS15b] B. Cogliati and Y. Seurin. On the Provable Security of the Iterated Even-Mansour Cipher Against Related-Key and Chosen-Key Attacks. In *EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 584–613. Springer, 2015.

[DFJ13] P. Derbez, P.A. Fouque, and J. Jean. Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting. In *EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 371–387. Springer, 2013.

[Din15] I. Dinur. Cryptanalytic Time-Memory-Data Tradeoffs for FX-Constructions with Applications to PRINCE and PRIDE. In *EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 231–253. Springer, 2015.

[DK13] O. Dunkelman and N. Keller. Cryptanalysis of the Stream Cipher LEX. *Designs, Codes and Cryptography*, 67(3):357–373, 2013.

[DKS12] O. Dunkelman, N. Keller, and A. Shamir. Minimalism in Cryptography: The Even-Mansour Scheme Revisited. In *EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 336–354. Springer, 2012.

[DR02] J. Daemen and V. Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[Dwo05] M. Dworkin. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST special publication 800-38b, http://csrc.nist.gov/publications/nistpubs/800-38B/SP800-38B.pdf. 2005.

[Dwo07] M. Dworkin. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf. 2007.

[EM97] S. Even and Y. Mansour. A Construction of a Cipher from a Single Pseudorandom Permutation. *Journal of Cryptology*, 10(3):151–162, 1997.

[FJM14] P.A. Fouque, A. Joux, and C. Mavromati. Multi-user Collisions: Applications to Discrete Logarithm, Even-Mansour and PRINCE. In *ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 420–438. Springer, 2014.

[FJP04] P.A. Fouque, A. Joux, and G. Poupard. Blockwise Adversarial Model for On-line Ciphers and Symmetric Encryption Schemes. In *SAC 2004*, volume 3357 of *Lecture Notes in Computer Science*, pages 212–226. Springer, 2004.

[FP15] P. Farshim and G. Procter. The Related-Key Security of Iterated Even-Mansour Ciphers. In *FSE 2015*, volume 9054 of *Lecture Notes in Computer Science*, pages 342–363. Springer, 2015.

[GJMN16] R. Granger, P. Jovanovic, B. Mennink, and S. Neves. Improved Masking for Tweakable Blockciphers with Applications to Authenticated Encryption. In *EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 263–293. Springer, 2016.

[GPV15] C. Garman, K.G. Paterson, and T. Van der Merwe. Attacks Only Get Better: Password Recovery Attacks Against RC4 in TLS. In *24th USENIX Security Symposium*, pages 113–128. USENIX Association, 2015.

[Hel80] M.E. Hellman. A Cryptanalytic Time-Memory Trade-Off. *IEEE Transactions on Information Theory*, 26(4):401–406, 1980.

[JMV02] A. Joux, G. Martinet, and F. Valette. Blockwise-Adaptive Attackers: Revisiting the (In)Security of Some Provably Secure Encryption Models: CBC, GEM, IACBC. In *CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 17–30. Springer, 2002.

[JNP14] J. Jean, I. Nikolic, and T. Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In *ASIACRYPT, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288. Springer, 2014.

[KR11] T. Krovetz and P. Rogaway. The Software Performance of Authenticated-Encryption Modes. In *FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327. Springer, 2011.

[LRW11] M. Liskov, R. Rivest, and D. Wagner. Tweakable Block Ciphers. *Journal of Cryptology*, 24(3):588–613, 2011.

[Mav15] C. Mavromati. Key-Recovery Attacks Against the MAC Algorithm Chaskey. In *SAC 2015*, volume 9566 of *Lecture Notes in Computer Science*, pages 205–216. Springer, 2015.

[Men12] A. Menezes. Another Look at Provable Security. In *EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, page 8. Springer, 2012.

[ML15] N. Mouha and A. Luykx. Multi-key Security: The Even-Mansour Construction Revisited. In *CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2015.

[MOV96] A. Menezes, P. Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[MS01] I. Mantin and A. Shamir. A Practical Attack on Broadcast RC4. In *FSE 2001*, volume 2355 of *Lecture Notes in Computer Science*, pages 152–164. Springer, 2001.

[PPS14] K. Paterson, B. Poettering, and J. Schuldt. Plaintext Recovery Attacks Against WPA/TKIP. In *FSE 2014*, volume 8540 of *Lecture Notes in Computer Science*, pages 325–349. Springer, 2014.

[RFC05] RFC. Internet Key Exchange (IKEv2) Protocol. https://tools.ietf.org/html/rfc4306. 2005.

[Rog04] P. Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In *ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.

[SRQL02] F. Standaert, G. Rouvroy, J. Quisquater, and J. Legat. A Time-Memory Tradeoff Using Distinguished Points: New Analysis & FPGA Results. In *CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 593–609. Springer, 2002.

[STA+15] Y. Sasaki, Y. Todo, K. Aoki, Y. Naito, T. Sugawara, Y. Murakami, M. Matsui, and S. Hirose. Minalpher v1.1. CAESAR (2015). https://competitions.cr.yp.to/round2/minalpherv11.pdf. 2015.