On the Security of the XOR Sandwiching Paradigm for Multiple Keyed Block Ciphers

Ruth Ng Ii-Yung [1], Khoongming Khoo[2] and Raphael C.-W. Phan [3]
[1]*The College, University of Chicago, 5801 S. Ellis Avenue, Chicago IL 60637*
(A part of this research was done while the author was at DSO National Laboratories)
[2]*DSO National Laboratories, 20 Science Park Drive, S118230, Singapore*
[3]*Faculty of Engineering, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia.*
ruthfrancisng@uchicago.edu, kkhoongm@dso.org.sg, raphael@mmu.edu.my

Keywords: Data-Encryption Standard; Block Ciphers; Meet-in-the-Middle; Related-Key

Abstract: While block cipher design is relatively mature, advances in computational power mean that the keylength of block ciphers, upon which the security relies entirely, becomes less resistant to cryptanalysis over time. Therefore, the security for a block cipher with a particular keylength typically is seen to last for at most some decades. One common approach to strengthen a block cipher's security is based on increasing its keylength. In the literature, two strategies have emerged: multiple keyed multiple encryption and multiple keyed XOR sandwiching. Known attacks on these such as Meet-in-the-Middle(Merkle and Hellman, 1981; van Oorschot and Wiener, 1991; Lucks, 1998) and Related-Key (J. Kelsey and Wagner, 1996; Choi et al., 1996; Vaudenay, 2011; Phan, 2004) attacks, show that Triple Encryption is significantly weaker than a brute-force attack would suggest, especially for block ciphers with small keys, such as the Data Encryption Standard (DES). This paper provides a comprehensive analysis on the security of the XOR sandwiching paradigm against known attacks for the case of multiple keyed triple encryption, w.l.o.g. using DES as the underlying block cipher. In particular, we focus on DES-XEXEXEX variants, based on 2-Key and 3-Key Triple-DES, which involve performing the XOR for key-whitening before and after each encryption with an additional 64-bit key. One of the conclusions to be drawn from this work is the increased strength obtained from the XOR sandwiching paradigm while requiring little in terms of additional computational resources.

1 INTRODUCTION

Work on the Data Encryption Standard (DES) in the areas of Meet-in-the-Middle Attacks and Related-Key Attacks have revealed 2-Key and 3-Key Triple DES to be much weaker than a naïve attack would suggest. We therefore hope to strengthen such encryption by increasing key-length.

Perhaps the most obvious response would be to increase these Triple Encryption DES variants to Quadruple Encryption DES variants. However, a quick calculation by a traditional Meet-in-the-Middle attack will reveal that both Triple-DES and Quadruple-DES can be attacked with a time complexity of 2^{113}, an ominous sign suggesting that the extra computational time of the added DES encryption is both needlessly cumbersome and insufficient to increase security.

1.1 Our Contribution

What we propose is to use an XOR-sandwiching paradigm to include an additional 64-bit key into a multiple encryption scheme. Specifically, we propose an XEXEXEX model (Figure 1) as an extension to both 2-Key-Triple-Encryption and 3-Key-Triple-Encryption, by XORing an additional 64-bit key in between each encryption call. These are also easy to implement in existing triple-encryption systems. As in DES-EXE and DES-X, the use of the XOR function for key-whitening strengthens the encryption scheme with negligible computational overhead.

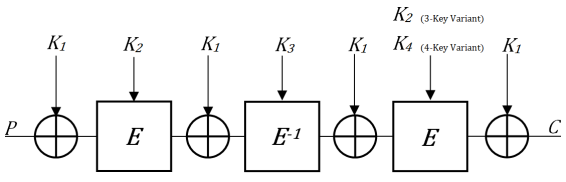

Figure 1: DES-XEXEXEX Variants Proposed

We present recent attacks to justify the choice of such an encryption scheme. As far as we know, major steps in breaking Triple-Encryption include the basic Meet-in-the-Middle (MITM) attack and its optimization (Lucks, 1998), MITM variants that targets 2-key triple-DES (Merkle and Hellman, 1981; van Oorschot and Wiener, 1991). In addition, we study Related-Key (RK) MITM attacks that exploits key differences (J. Kelsey and Wagner, 1996; Choi et al., 1996) and RK-MITM attacks that exploits key-permutation (Phan, 2004; Vaudenay, 2011). In this research, we argue that our XEXEXEX encryption variant significantly strengthens Triple-Encryption against known attacks, through the example of DES.

2 MEET-IN-THE-MIDDLE ATTACK

The traditional MITM attack is described diagrammatically (Figure 2) below, comparing the original Triple-Encryption with the one which we propose.

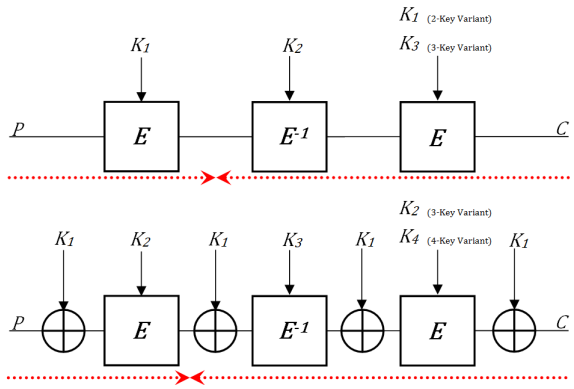

Figure 2: Traditional MITM Attack on Triple-DES and DES-XEXEXEX Variants

For 3-Key Triple-DES, we obtain (P,C), a known Plaintext-Ciphertext (PT-CT) pair, and consider the possible K_1 separately from possible (K_2,K_3), seeking $E_{K_1}(P) = E_{K_2}(E_{K_3}^{-1}(C))$. Note that we accept values of (K_1,K_2,K_3) satisfying the above equation if the encryption is true for $\lceil \log_{64} 2^{168} \rceil = 3$ PT-CT pairs. Notice that this attack requires a time complexity in the order of 2^{113} encryptions and a memory complexity of $(64+56)2^{56} \approx 2^{63}$ bits.

A very similar search for $E_{K_2}(P \oplus K_1) \oplus K_1 = E_{K_3}(E_{K_4}^{-1}(C \oplus K_1) \oplus K_1)$ can be carried out for our 4-Key variant. However, to remove significant memory complexity we consider this attack individually for each value of K_1 since that is constant in the encryption scheme. We accept values of (K_1,K_2,K_3,K_4) if the results are consistent over $\lceil \log_{64} 2^{212} \rceil = 4$ PT-CT pairs. This attack will have a time complexity in the order of 2^{177} and a memory complexity of approximately 2^{63}.

As for 2-Key Triple DES, through a similar logic as suggested above, we consider each value of K_1 separately. We expect that $\lceil \log_{64} 2^{112} \rceil = 2$ known PT-CT pairs will confirm the correct value of K_1 and K_2 with a time complexity of 2^{113} and a negligible memory requirement. The logical extension will mean that for our 3-Key variant we consider each (K_1,K_2) individually, and accept values that are consistent over $\lceil \log_{64} 2^{176} \rceil = 3$ PT-CT values. We arrive at a time complexity of 2^{177} and a negligible memory complexity.

The addition of an additional key in the proposed XEXEXEX model has thus increases the time complexity of a basic MITM attack by 2^{64}, an identical increase to what we would expect from a naïve attack.

2.1 Merkle-Hellman MITM Attack

The Merkle-Hellman MITM attack (Merkle and Hellman, 1981) is a chosen-plaintext alternative to this. The common application of this attack is in the case of 2-Key Triple Encryption (Figure 3). In 2-Key Triple-DES, we decrypt some 64-bit value A based on all 2^{56} possible values of K_1. For each A, we make a chosen plaintext encryption query to obtain the corresponding ciphertext and decrypt each A via the guessed K_1 as before. We then store these values and exhaustively search all K_2 such that $E_{K_1}^{-1}(Enc(E_{K_1}^{-1}(A))) = E_{K_2}^{-1}(A)$ (where Enc is the chosen plaintext encryption query). We accept a value of (K_1,K_2) when $\lceil \log_{64} 2^{112} \rceil = 2$ PT-CT pairs are consistent with those keys. This attack has a time complexity of $3(2^{56}) \approx 2^{57.6}$ Encryptions (neglecting that of obtaining the ciphertexts of 2^{56} chosen-PT) and a memory complexity of 2^{63}.

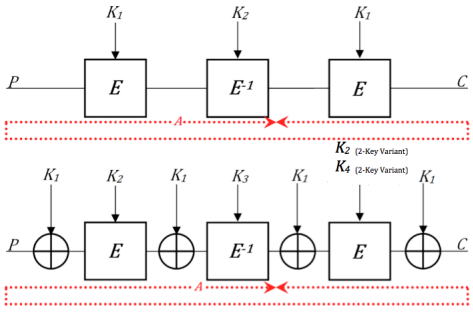

Figure 3: Merkle-Hellman MITM Attacks on 2-Key Triple-DES and both DES-XEXEXEX variants.

As for our 3-Key variant of the above attack,

we consider combinations of (K_1, K_3) separately from K_2, as represented diagrammatically above (Figure 3). Specifically, the equality we search for is $E_{K_2}^{-1}(Enc(E_{K_2}^{-1}(A \oplus K1) \oplus K1) \oplus K1 = E_{K_3}^{-1}(A)$. We accept a value of (K_1, K_2, K_3) when $\lceil \log_{2^{64}} 2^{176} \rceil = 3$ PT-CT pairs are consistent with those keys. This has a time complexity of $2^{121.6}$ and a memory complexity of 2^{63}. This also requires the entire codebook of PT-CT pairs.

For the original 3-Key Triple-DES and 4-Key DES-XEXEXEX algorithm, this gives us no advantage over the original MITM attack. Note that while the attack, with the entire codebook of PT-CT pairs, we can consider possible K_2 separately from the remaining keys, this gives negligible time advantage.

Therefore, similar to the original MITM attack, our variant of 2-Key Triple DES has succeeded in increasing the complexity of a chosen plaintext MITM attack by a factor of 2^{64}.

2.2 Van Oorshot - Wiener MITM Attack

Van Oorshot and Wiener's proposal to extend Merkle-Hellman's chosen plaintext attack to a known-plaintext attack is applicable to the case of 2-Key Triple DES, where the Merkle-Hellman attack gives us a significant reduction in complexity on the original MITM attack (van Oorschot and Wiener, 1991). We choose 2^{32} values of P. For each P, we calculate all 2^{56} possible values of $E_{K_1}^{-1}(P)$ and check these against the 2^{32} PT-CT pairs. For the matches we find, we compute $B = E_{K_1}^{-1}(C)$ and store (K_1, B) using at most 2^{56} memory entries. On each of these, we conduct an exhaustive search of K_2 and test resultant candidate (K_1, K_2) pairs with additional PT-CT pairs. We repeat this process for different values of P until the correct key is found. With 2^{32} known PT-CT pairs, this attack has time complexity of 2^{89} encryptions and a memory complexity of $(64+56)2^{56} \approx 2^{63}$.

Similarly, we apply this to the Merkle-Hellman attack on our 3-key DES-XEXEXEX variant as described in Section 2.1. Starting with 2^{32} PT-CT pairs, we accept a value of (K_1, K_2, K_3) when $\lceil \log_{2^{64}} 2^{176} \rceil = 3$ PT-CT pairs are consistent with those keys. This attack expects a time-complexity of 2^{153} and memory-complexity of 2^{63}.

Therefore, in the case of 2-Key Triple DES and its variant, we have shown that the time complexity increase in the implementation of the XEXEXEX variant is 2^{63}, similar to the attacks discussed above.

2.3 Lucks MITM Attack

As for 3-Key Triple DES, Lucks proposes an optimization which reduces the time complexity with increased memory (Lucks, 1998). He presents a variety of attacks, however, we select the attack with comparable requirements to other attacks we present and which considers DES as an ideal cipher, for fair comparison. His most efficient attack involves a set 2^{32} PT-CT pairs $(p_1, c_1), \ldots (p_{2^{32}}, c_{2^{32}})$ and a second set $S \subset \{0,1\}^{64}$ and $|S| = 2^{33}$. Due to the complexity of his attack, we paraphrase his attack below:

1. For all $a \in S$, we define the sets $M_a = \{(i, K_1) \in \{1, \ldots 2^{32}\} \times \{0,1\}^{56} \mid E_{K_1}(p_i) = a\}$, which can be computed with complexity $2^{32} \times 2^{56} = 2^{88}$ and stored in memory.
2. For all $b \in \{0,1\}^{64}$ and $i \in \{1, \ldots 2^{32}\}$, we define the sets $N_{b,i} = \{K_3 \in \{0,1\}^k \mid E_{K_3}(b) = c_i\}$. They can be computed with complexity $2^{32} \times 2^{56} = 2^{88}$, by computing $b = E_{K3}^{-1}(c_i)$ for all 2^{32} c_i and for all 2^{56} K_3, and placing K_3 into the corresponding set $N_{b,i}$, which is stored in memory.
3. For all $K_2 \in \{0,1\}^{56}$ and $a \in S$ we search for some $N_{b,i}$ such that $E_{K_2}^{-1}(a) = b$. Then, for such (K_2, a, b) where $(i, K_1) \in M_a$ and $K_3 \in N_{b,i}$ we enter (K_1, K_2, K_3) into a hash table.
4. When some triple (K_1, K_2, K_3) is entered a second time into the hash table, we test the set of keys with 1 other values of (p_i, c_i). Notice that this is sufficient since we accept a value of (K_1, K_2, K_3) when $\lceil \log_{2^{64}} 2^{168} \rceil = 3$ PT-CT pairs are consistent with those keys.

We refer the reader to (Lucks, 1998) for detailed calculations to derive the requirements of the attack. 2^{33} values of $a \in S$, 2^{32} PT-CT pairs, 2^{88} encryptions and $2^{88}(56) \approx 2^{93.8}$ memory-complexity are required.

We then attempt to apply this to our 4-Key DES-XEXEXEX variant. We considered two possible methods of adapting the attack. The first is to simply repeat the attack by guessing values of K_1, and repeating this for all values of K_1. This would mean that the time complexity would simply be $2^{90} \cdot 2^{64} = 2^{154}$ single encryptions and memory complexity, reusable for each K_1, will be $2^{93.8}$ bits. Note, though, that as $\lceil \log_{2^{64}} 2^{176} \rceil = 4$, "tripletest" will now need to test the candidate keys on two additional PT-CT pairs.

The second method involved, for an arbitrary (P,C) pair, define $a = E_{K_2}(p \oplus K_1)$ and $b = E_{K_4}^{-1}(c \oplus K_1)$. We then calculate M_{a,K_1} and N_{b,i,K_1}, sets identical to that which we studied before, but restricted to each K_1. The rest of the attack proceeds by considering each value of K_1 individually then searching

for values to "tripletest" (searching for $E_{K_3}^{-1}(a \oplus K_1) = b \oplus K_1$). However, this would come at the cost of 2^{64} times more memory and would wind down to a comparable time complexity because each K_1 would still be considered as an individual case. Therefore, with the first method preferable, we can once again report an additional 2^{64} increase in time complexity of an attack with the addition of the 64-bit K_1 key.

Note that the Lucks' attack is inefficient when the first and third DES encryption make use of the same key since much recalculation would be done. To this end, the Merkle-Hellman or Van Oorshot-Wiener attack is much more efficient. Therefore, we did not consider an application of Lucks' attack on 2-Key Triple-DES and our 3-Key variant of it as part of our study.

3 RELATED-KEY ATTACKS

We also consider their security under Related-Key attacks, something which is posited to be not as purely theoretical as it seems in recent years (Phan, 2004).

3.1 Kelsey-Wagner-Schneier Related-Key Attack

We begin with the original Kelsey-Wagner-Schneier Related-Key Attack (J. Kelsey and Wagner, 1996). This attack on Triple-DES involves a known PT-CT pair, (P,C) encrypted on unknown keys (K_1,K_2,K_3) and the resultant ciphertext being decrypted under keys $(K_1 \oplus \Delta, K_2, K_3)$, where Δ is known, to arrive at P'. Then, a exhaustive search can be done for K_1 since $E_{K_1}(P) = E_{K_1 \oplus \Delta}(P')$. From here, a MITM attack can be performed on the remaining two keys, similar to that which is performed on double-DES. Notice that this will require approximately $3(2^{56}) \approx 2^{57.6}$ encryptions and 2^{63} memory complexity. Note that since $\lceil \log_{2^{64}} 2^{168} \rceil = 3$, we would also need to test resultant pairs against 2 other known PT-CT values.

A similar attack can be arranged for 4-Key DES-XEXEXEX, given related keys (K_1,K_2,K_3,K_4) and $(K_1 \oplus \Delta, K_2, K_3, K_4)$. A known PT-CT pair is encrypted on the former and the resultant ciphertext decrypted on the latter. This allows us to do a exhaustive search on combinations of (K_1,K_2), and conduct a MITM attack to find possible (K_3,K_4) for each candidate (K_1,K_2). Notice that we will accept a combination of keys if it is consistent over $\lceil \log_{2^{64}} 2^{232} \rceil = 4$ PT-CT pairs. We expect a time complexity of 2^{121} encryptions and a memory complexity of $2^{56}(56+64) = 63$. This attack is not applicable to 2-Key Triple-DES and our 3-Key DES-XEXEXEX variant.

3.2 Choi et al. Related-Key Attack

Given that a chosen-plaintext attack is considered unfeasible at present (van Oorschot and Wiener, 1991), a chosen-CT attack is even less useful. In this regard, most studies look to the known PT-CT attack presented by Choi et al (Choi et al., 1996) as diagrammatically represented below (Figure 4).

With 2^{32} known PT-CT pairs encrypted under the keys and another 2^{32} known PT-CT pairs encrypted under the Related-Keys, we search for collisions as indicated by the arrows. For 2-Key Triple DES, we search for $(P,C),(P',C')$ and K_1 such that $E_{K_1}(P) = E_{K_1 \oplus \Delta}(P')$ and $E_{K_1}^{-1}(C) = E_{K_1 \oplus \Delta}^{-1}(C')$ are both satisfied. For 3-Key Triple DES, we search for $(P,C),(P',C')$ and K_1 such that $E_{K_1}(P) = E_{K_1 \oplus \Delta}(P')$ and $C = C'$. We expect to exist by the Birthday Paradox. With these candidates, we do a MITM search for the remaining keys. We will accept the keys when they are consistent across $\lceil \log_{2^{64}} 2^{168} \rceil = 3$ PT-CT pairs for Three-Key Triple-DES and $\lceil \log_{2^{64}} 2^{112} \rceil = 2$ PT-CT pairs for Two-Key Triple-DES.

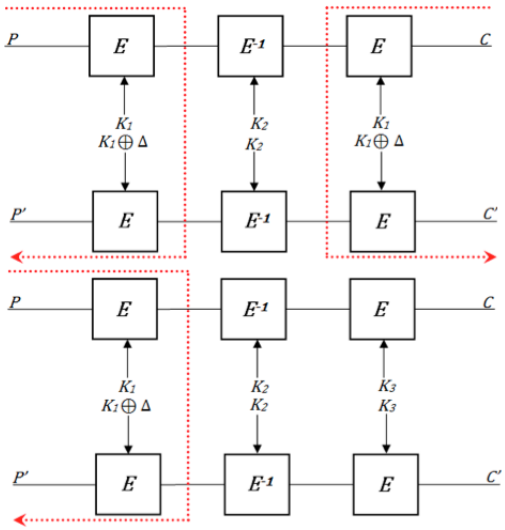

Figure 4: RK attacks on 2-Key Triple DES and 3-Key Triple DES

This has an expected time complexity of 2^{89} encryptions and a memory complexity of 2^{39} for 2-Key Triple-DES. For 3-Key Triple-DES, this is a time complexity of $2^{57.6}$ single encryptions and 2^{60} of memory complexity (Choi et al., 1996).

Extending this to our XEXEXEX variants, we make use of the key-relation (K_1,K_2,K_3,K_4) and $(K_1,K_2 \oplus \Delta, K_3, K_4)$ or (K_1,K_2,K_3) and $(K_1,K_2 \oplus$

$\Delta, K_3)$. As before, there is the solution of simply repeating the entire attack for all guesses of K_1, modifying the collision search to include the relevant XOR functions. This will leave us with identical memory complexity for both attacks and a time complexity of 2^{153} encryptions for 3-Key DES-XEXEXEX and 2^{121} for 4-Key DES-XEXEXEX. Notice, however, that we would only accept a set of keys after $\lceil \log_{2^{64}} 2^{212} \rceil = 4$ PT-CT pairs are consistent with the results. However, we present an alternative here, which might occur as a logical extension to a reader (Figure 5).

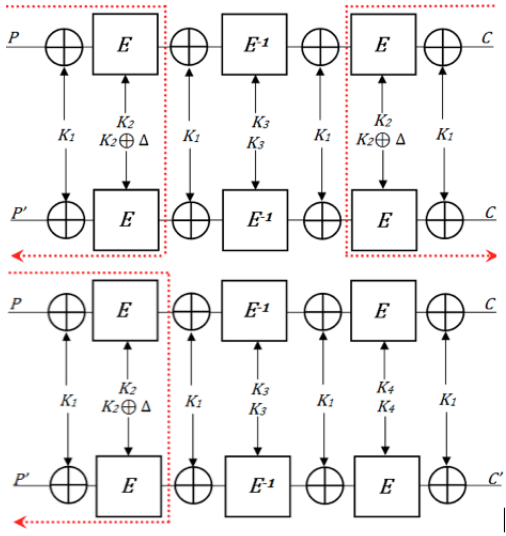

Figure 5: Alternate RK attacks on DES-XEXEXEX variants

For both DES-XEXEXEX variants, we search among 2^{32} known PT-CT pairs encrypted under keys (K_1, K_2, K_3) or (K_1, K_2, K_3, K_4) and their Related-Keys, $(K_1, K_2 \oplus \Delta, K_3)$ and $(K_1, K_2 \oplus \Delta, K_3, K_4)$. We instead guess all (K_1, K_2) and search for a (P, C) pair such that $E_{K_2}(P \oplus K_1) = E_{K_2 \oplus \Delta}(P' \oplus K_1)$ and $E^{-1}_{K_2}(C \oplus K_1) = E^{-1}_{K_2 \oplus \Delta}(C' \oplus K_1)$, in the case of 3-Key DES-XEXEXEX, and that $E_{K_2}(P \oplus K_1) = E_{K_2 \oplus \Delta}(P' \oplus K_1)$ and $C = C'$, in the case of 4-Key DES-XEXEXEX. We then do a exhaustive search for the remaining keys. However, this wastefully computes and stores all encryptions of all possibilities of K_1, in memory, without performing better than the first method.

3.3 Vaudenay RK Attack

The RK attack proposed by Vaudenay (Vaudenay, 2011) on Three-Key Triple-DES notes that if we were to encrypt a plaintext, P, according to keys (K_1, K_2, K_3) then decrypt the ciphertext according to Related-Keys $\phi(K_1, K_2, K_3) = (K_2, K_1, K_3)$ to give a second ciphertext, C. This allows us to yield the following relation: $(E_{K_1} \circ E^{-1}_{K_2})^2(P) = C$. From this, we streamline a list of plaintexts, x, which yield "fixed points" where $(E^{-1}_{K_2}(E_{K_1}(x)) = x$ under the correct keys, K_1, K_2. With candidate K_1, K_2 proposed, and the respective x, we can then do a exhaustive search for K_3 and test the result on more PT-CT pairs.

Vaudenay presents an attack based on known PT-CT pairs and another based on Broadcast Known Plaintexts (BKP). However, since we are more interested in comparing Triple-DES to our variants, and not so much on comparing the results of various attacks, we will study the BKP variant and acknowledge that our results can be trivially adapted to the known Plaintext variant of Vaudenay's attack.

We refer the reader to Vaudenay's report (Vaudenay, 2011) for the exact procedure of Vaudenay's rather complex attack, as well as his in-depth calculations of complexity. For 3-Key Triple-DES, he proposes to let n, the number of pairs of Related-Keys considered to be 3 (meaning we will have known PT-CT pairs encrypted under $K, \phi(K), K \oplus \Delta_1, \phi(K \oplus \Delta_1), K \oplus \Delta_2, \phi(K \oplus \Delta_2)$ where Δ_1, Δ_2 are known. To this end, we have the number of BKP required to be 2^{67} and R_n, the expected number of wrong keys that are considered in the second part of the attack, is approximately $2^{-1.72}$ and this yields an expected time complexity of $2^{57} \cdot 3 \approx 2^{58.6}$ and an expected memory complexity of approximately 2^{63}.

As for Two-Key Triple-DES, we instead consider Encryption of some plaintext P by keys (K_1, K_2) and decryption of the ciphertext by Related-Keys (K_2, K_1) to give C. This yields the equation $(E_{K_1} \circ E^{-1}_{K_2})^3(P) = C$. Similarly, fixed points of the same form as above are sieved out. However, wrong key-guesses are easily discarded by a consistency check, meaning that we we can have $n = 1$. This requires 2^{65} BKP and yields a time complexity of $2^{57.6}$ single encryptions and a memory complexity of 2^{63}.

The purpose of finding fixed points in this attack is to be able to consider the behaviour of a subset of the keys. In this case, it is that of K_3, by requiring that the plaintext which enters the encryption scheme has a high-chance of being identical to the ciphertext before it is encrypted by K_3. In the case of 2-Key Triple DES, the same thing is achieved for the last K_1. This is done by exploiting the second DES function in the encryption scheme being a decryption and the specific key relation. Notice, however, that in an XEXEXEX variant, if we hold K_1 constant across the Key-Relation, regardless of whether we perform iterations of encryptions, decryptions or some combination of the two, independent of the permutation of other keys, the identical XOR function performed after the triple-

encryption and the start of the second triple encryption will cancel out. However, the XOR functions between the encryptions are not affected. This means that the resultant function will not be repeating in the way that we were able to achieve in Vaudenay's attack since the encryptions can no longer take on a consistent pattern.

However, consider instead 2^{32} $(P, C \oplus K_1)$ values, for some guessed K_1. We can then attack 3-Key DES-XEXEXE by guessing all possible K_1. Similarly, by guessing K_1, we can compute 2^{32} $(P \oplus K_1)$ values for each K_1 and attack 4-Key DES-EXEXEX. To this end, notice that now we can perform the same combination, of an encryption and decryption, on P to arrive at C, with Related Keys (K_1, K_2, K_3) and (K_1, K_3, K_2) for the 3-Key Variant as well as for (K_1, K_2, K_3, K_4) and (K_1, K_3, K_2, K_4) for the 4-Key variant. Define a function, $x \oplus K_1 = f(x)$, then, we can arrive at equations similar to those above where $(f \circ E_{K_2} \circ f \circ E_{K_3}^{-1})^3(P) = C$ for the 3-Key Variant and $(f \circ E_{K_2} \circ f \circ E_{K_3}^{-1})^2(P) = C$ for the 4-Key Variant.

Notice then that we need to change the value of n (number of pairs of K, $\phi(K)$ we consider) since keylength has been increased, in 4-Key DES-EXEXEX. We therefore have the following calculations (adapted from (Vaudenay, 2011), Section 3.1, pg 5-6):

First, we calculate the expected f n^*, the number of lists with an odd number of fixed points. Let $n = 6$,

$$E(n^*) = 1 + (n-1)\frac{1 - e^{-\frac{3}{2}}}{2} \approx 2.94 \qquad (1)$$

Then, we have that there are $2^{2(56)+(64)}$ possible combinations of keys but an equation to satisfy on $(2.94)(64)$ bits. This gives us the respective value of R_n (expected number of wrong keys in R given n Related-Key pairs) as:

$$R_n \approx 2^{2(56)+(64)-(2.94)(64)} = 2^{-12.16} \qquad (2)$$

Notice that the value of n does not impact the choice to repeat the entire attack 2 times (i.e. N_n is unrelated to n so long as $a > 0$). Therefore, with an identical success rate, we require $6(2^{64+1}) \approx 2^{68}$ BKP. The only difference in the time complexity which sees a 2^{64} increase in the calculations to arrive at a fixed point, since each K_1 must be guessed separately. Note that the XOR functions to derive each set of $(P, C \oplus K_1)$ or $(P \oplus K_1, C)$ values are assumed to be of negligible complexity. This has a time complexity of $2(2 \cdot 2^{56+64} + 2^{56} \cdot 2^{-12.16}) + 2^{56} \approx 2^{122}$ encryptions. Memory can be reused for each guess of K_1, therefore, we have that the memory complexity is 2^{63}.

For the Three-Key DES-XEXEXE, we can adopt the same method of finding fixed points, however, as in the original attack on Two-Key Triple-DES, take $n = 1$. This has time complexity of $4(2^{56+64}) = 2^{122}$ encryptions and a memory complexity of 2^{63}.

Notice that in both these cases, the time complexities are comparable to that of the attack we considered on DES-XEXEXEX. This, we realized, is because Vaudenay's attacks on DES-EXEXEX and DES-XEXEXE involve guessing each K_1 in turn, returning to an attack very reminiscent of that of DES-XEXEXEX. This makes the memory space 2^{64} times less, and reduces the required value of n, other indicators that the attack is identical in nature. Therefore, we have shown the robustness of our method of strengthening Triple-DES, in that, even if RK attacks such as Vaudenay's attack could be more than trivially applied, we still achieve a 2^{64} complexity increase for the 64-bits of added keylength.

3.4 Phan RK Attack

Phan's RK slide-attack can be applied to both the 2-Key and the 3-Key Triple Encryption effectively, as discussed in his paper (Phan, 2004). We refer the reader to his paper for the exact details of each attack.

With 2^{32} PT-CT pairs each for the original key and the Related-Key, we can expect 1 pair with the desired relation by the Birthday Paradox. The first set of encryptions (for all possible K_1 on all values of P) dominates the time complexity, meaning that $2^{56} \cdot 2^{32} = 2^{88}$ single-DES encryptions are required for the attack. The memory complexity is also dominated by this step, $2^{88} \cdot (56 + 64 + 64) \approx 2^{96}$.

In the 3-Key Triple DES, we consider PT-CT pairs encrypted under the keys (K_1, K_2, K_3) and (K_2, K_3, K_1). We then search for (P, C), encrypted under (K_1, K_2, K_3) and (P', C') encrypted under (K_2, K_3, K_1) such that $C' = E_{K_1}(P)$ and $C = E_{K_1}(P')$. Once again, we obtain 2^{32} PT-CT pairs for each set of keys and create a list of candidates for K_1 by encrypting each P and decrypting each C according to each K_1. Those satisfying the collision conditions give candidate values for K_1. This, as he reports, requires 2^{88} DES encryptions and a memory complexity of $2^{32} \cdot (64 + 64) = 2^{39}$.

Notice that an exhaustive search for K_2, K_3 via a traditional MITM attack applies once K_1 has been determined. This can be achieved with 2^{39} memory complexity by portioning the 2^{56} candidates for K_1 into sets of 2^{32} values, a separate MITM attack is then performed using an exhaustive key-search for K_2 and matching against possible ciphertext values given for each group. The total time complexity of this search should be$2^{(56-32)} \cdot (2^{56}) = 2^{80}$, negligible in comparison to the time-complexity of the main attack.

Similar to our analysis in other sections of this pa-

per, to achieve a time-complexity lower than 2^{64} times that of the original attack, the attack must segment the keys into two mutually exclusive groups. Notice that should be attempt a slide attack on either DES-XEXEXEX variant, to isolate one or more encryptions, XOR functions both inside and outside the shared segment of the encryption scheme encryption under the pair of Related-Keys must follow, making them not mutually exclusive. Therefore, our focus turns to the search for a method for the Phan attack to achieve this complexity.

For this, we obtain 2 sets of 2^{32} PT-CT pairs, encrypted under (K_1, K_2, K_3, K_4), denoted (P^*, C^*), and (K_1, K_3, K_2, K_4), denoted (P'^*, C'^*). We guess a particular K_1 and XOR all 2^{33} PT-CT pairs by it, to arrive at $(P,C) = (P^* \oplus K_1, C^* \oplus K_1)$ and $(P', C') = (P'^* \oplus K_1, C'^* \oplus K_1)$, which is of negligible time-complexity. Then, the attack can proceed as diagrammatically displayed below (Figure 6). This yields an identical memory complexity and a time complexity of 2^{144} encryptions for 4-Key DES-XEXEXEX and 2^{152} encryptions for 3-Key DES-XEXEXEX.

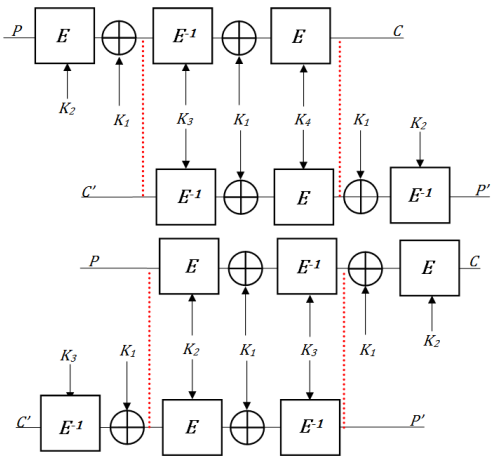

Figure 6: Phan attacks on 4-Key DES-EXEXEX and 3-Key DES-XEXEXE

4 Conclusion

The merits of the 3-Key and 4-Key DES-XEXEXEX variants in strengthening Triple-DES against known MITM and RK attacks have been extensively shown. Due to the fact that we employ the XOR function, using the same key, across the entire encryption scheme, we have arrived at a cipher which cannot readily be portioned into segments with independent keys to be attacked separately. Therefore, the additional key will be effective in strengthening the cipher as opposed to other uses of an additional key in the cipher. This is shown in our research where all the known attacks we presented reports an additional complexity of about 2^{64} times for a key-extension of 64 bits. In addition, the XOR function involves negligible computation, thereby not affecting the implementation of the cipher. 3-Key or 4-Key Quadruple-DES is an example of an intuitive solution that does not satisfy these conditions.

Therefore, we believe that our contribution is useful to systems still employing multiple-encryption structures with insufficient security afforded by its key-length, though additional key-bits can be sought at reasonable cost. Beyond DES, similar results will be obtained in application to any block cipher, meaning, for a cipher with block-size n, we achieve a 2^n increase in security given a n-bit key-length increase.

4.1 Possible Future Extensions

Recent literature, such as (Phan, 2004) and (Kilian and Rogaway, 1996) support moving away from XOR to addition modulo 64 (+), with the belief that it is stronger. Intuitively, the ability for "bit-carrying" given by addition reduces the susceptibility to attacks involving "weak keys". Also, we have that the inverse function of XOR is itself, whereas addition modulo is not that symmetrical, invalidating some attacks. However, Phan presents an attack that is applicable to DES-+ and not DES-X (Phan, 2004). We briefly studied the DES - +E+E+E+ model for Triple-Encryption, with similar results to DES-XEXEXEX. The only exception was the Vaudenay attack, which we will not be able to carry out at all since the encryption and decryption schemes will be completely different. Therefore, they cannot compose a repeating function by which fixed points can be found. Future work can study addition in relation to this in more detail.

Also, we considered a general $t+1$-key DES-$(XE)^tX$ encryption scheme and we believe that similar attacks can be applied to show that a 2^{64} increase in security is achieved. However, more research can be done on this to study the significance of this increase as t increases, as well as other schemes involving less or more than $t+1$ keys.

REFERENCES

Choi, J., Kim, J., Sung, J., Lee, S., and J.Lim (1996). Related-key and meet-in-the-middle attacks on triple-des and des-exe. In *Proceedings of the 2005 international conference on Computational Science and Its Applications - Volume Part II*. Springer-Verlag.

J. Kelsey, B. S. and Wagner, D. (1996). Key-schedule cryptoanalysis of idea, g-des, gost, safer, and triple-des. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*. Springer-Verlag.

Kilian, J. and Rogaway, P. (1996). How to protect des against exhaustive key search. In *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*. Springer-Verlag.

Lucks, S. (1998). Attacking triple encryption. In *Proceedings of the 5th International Workshop on Fast Software Encryption*. Springer-Verlag.

Merkle, R. and Hellman, M. (1981). On the security of multiple encryption.

Phan, R. (2004). Related-key and meet-in-the-middle attacks on triple-des and des-exe. In *In Topics in Cryptology - The Cryptographer's Track at RSA Conference (CT-RSA '04)*. Springer.

van Oorschot, C. and Wiener, M. (1991). Related-key attack against triple encryption based on fixed points. In *Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*. Springer-Verlag New York, Inc.

Vaudenay, S. (2011). Related-key attack against triple encryption based on fixed points. In *SECRYPT*. SCITEPRESS.

Table 1: Summary of Complexities for MITM and RK Attacks on Triple-Encryptions and DES-XEXEXEX variants

Encryption Scheme	Attack	PT-CT Pairs Requirement	Time (Encryptions)	Memory (bits)
3K Triple-DES	MITM	3 Known	2^{113}	2^{63}
3K Triple-DES	Lucks	2^{32} Known	2^{90}	$2^{93.8}$
3K Triple-DES	Kelsey et. al	1 Chosen-Decryption 2 Known	2^{57}	2^{63}
3K Triple-DES	Choi et. al	2^{33} RK-Known	2^{57}	2^{60}
3K Triple-DES	Vaudenay	2^{67} RK-BKP	$2^{58.6}$	2^{63}
3K Triple-DES	Phan	2^{33} RK-Known	2^{88}	2^{39}
4K DES-XEXEXEX	MITM	4 Known	2^{177}	2^{63}
4K DES-XEXEXEX	Lucks	2^{32} Known	2^{154}	$2^{93.8}$
4K DES-XEXEXEX	Kelsey et. al	1 Chosen-Decryption 2 Known	2^{121}	2^{63}
4K DES-XEXEXEX	Choi et. al	2^{33} RK-Known	2^{121}	2^{60}
4K DES-XEXEXEX	Vaudenay	2^{68} RK-BKP	2^{122}	2^{63}
4K DES-XEXEXEX (?)	Phan	2^{33} RK-Known	2^{152}	2^{63}
2K Triple-DES	MITM	2 Known	2^{113}	Negligible
2K Triple-DES	Merkle-Hellman	2^{56} Chosen	$2^{57.6}$	2^{63}
2K Triple-DES	Oorschot-Wiener	2^{32} Known	2^{89}	2^{63}
2K Triple-DES	Choi et. al	2^{33} 2^{33} RK-Known	2^{89}	2^{39}
2K Triple-DES	Vaudenay	2^{33} 2^{33} RK-Known	$2^{57.6}$	2^{63}
2K Triple-DES	Phan	2^{33} RK-Known	2^{88}	2^{96}
3K DES-XEXEXEX	MITM	3 Known	2^{177}	Negligible
3K DES-XEXEXEX	Merkle-Hellman	2^{120} Chosen	2^{117}	2^{63}
3K DES-XEXEXEX	Oorshot-Wiener	2^{33} Known	2^{153}	2^{63}
3K DES-XEXEXEX	Choi et. al	2^{33} RK-Known	2^{153}	2^{39}
3K DES-XEXEXEX	Vaudenay	2^{33} RK-Known	2^{121}	2^{63}
3K DES-XEXEXEX	Phan	2^{33} RK-Known	2^{152}	2^{96}