Breaking Full ChiLow-32

Jian Guo, Shichang Wang, and Tianyu Zhang

Nanyang Technological University, Singapore
{shichang.wang,guojian}@ntu.edu.sg, tianyu005@e.ntu.edu.sg

Abstract. ChiLow is a family of tweakable block ciphers proposed at EUROCRYPT 2025. In this paper, we present a cryptanalysis on ChiLow based on the Meet-in-the-Middle (MITM) attack framework. For ChiLow-32, we first present an MITM attack on full ChiLow-32 exploiting the cipher's diffusion properties, which achieves a time complexity of $2^{122.6}$ using 97 known plaintext-ciphertext (P-C) pairs. Building on this, we further introduce a refinement based on the linearization of χ function. By using more known pairs, we significantly improve the attack, reducing the time complexity to $2^{108.6}$ with 196 known P-C pairs. For ChiLow-40, we mount an attack on reduced-round versions: a 7-round attack with time complexity $2^{127.4}$ requiring 164 known P-C pairs, and a 6-round attack with time complexity $2^{88.9}$ requiring 162 known P-C pairs.

Keywords: Meet-in-the-Middle Attack · Key Recovery · ChiLow

1 Introduction

1.1 ChiChi and ChiLow

Lightweight cryptography has become increasingly important for applications in IoT, embedded systems, and other constrained environments, where limited energy, area, and latency impose strict design requirements. To address this need, NIST launched the Lightweight Cryptography (LWC) competition in 2019 with the goal of identifying efficient and secure algorithms suitable for widespread deployment. After four years of evaluation, Ascon was selected as the winner and subsequently standardized as NIST's lightweight cryptography standard for authenticated encryption and hashing. This milestone reflects the community's emphasis on achieving a careful balance between performance and security.

Beyond general-purpose lightweight algorithms, recent years have also witnessed the emergence of context-specific block ciphers tailored to specialized use cases. Instead of broad applicability, these schemes prioritize extreme latency, area, or block-size constraints arising in scenarios such as pointer protection or cache randomization. The recently proposed ChiLow [7] is a family of ultra-low-latency tweakable block ciphers (TBCs) and related pseudorandom functions (PRFs), presented at Eurocrypt 2025. ChiLow was designed to secure code stored in external memory for microprocessors, where each instruction must be

decrypted before execution. Its primary design goal is to minimize decryption latency, making it particularly relevant for such settings.

ChiLow employs a clean substitution–permutation network (SPN) structure, combined with a fast tweakey schedule. To achieve extreme performance, it uses a very small 32-bit block size and only 8 rounds. Implemented with the Nangate 15 nm Open Cell Library, the design achieves a decryption latency below 280 picoseconds. The proposal also introduces ChiChi, a non-linear layer derived from the well-known χ function, to further strengthen its security. The designers evaluated ChiLow against a range of attacks, including differential and linear cryptanalysis (with clustering effects), integral distinguishers, and a meet-in-the-middle key-recovery attack effective up to 4 rounds.

1.2 The Meet-In-The-Middle Attack

The meet-in-the-middle (MITM) attack is one of the most fundamental techniques in cryptanalysis. It dates back to Diffie and Hellman's classical attack on Double-DES [14], which showed how to partition a primitive and its search space into two separate parts, each explored independently. Later, Bogdanov and Rechberger formalized the three-subset MITM framework [10], providing a unifying view of earlier block-cipher attacks.

MITM techniques soon expanded beyond block ciphers. Sasaki and Aoki introduced MITM attacks against hash functions [28], work that later received the IACR Test-of-Time Award[1]. This line of research advanced quickly, with innovations such as splice-and-cut [2], initial structures [3,29], indirect and partial matching [1], probabilistic techniques [17], bicliques [23], and precomputations [11]. Sasaki's work at FSE 2011 gave the first preimage attacks on AES hashing [26], later refined by Bao *et al.* [4]. Subsequent progress came from automated frameworks, starting with Bao *et al.* 's search tool [5] and later expanded to key-recovery, collision, and more advanced structures [15,6,20,31,32]. Recently, these frameworks have also been applied to sponge constructions [25,16] and Feistel networks [19], and enriched with linearization, structural similarity [12], and same-color initial structures [13].

1.3 Contribution

While the designers' analyses provide useful initial insights, the reduced block size and round count naturally lead to tighter security margins. ChiLow represents a clean and well-motivated design that addresses an important problem in ultra-low-latency encryption, and its acceptance at a top-tier venue such as Eurocrypt underscores its relevance.

In this work, we combine the advanced meet-in-the-middle attack framework with cryptanalytic insights on the ChiLow structure, and present key recovery attacks on ChiLow-32 and ChiLow-40. Main results include two key recovery attacks on full round ChiLow-32 and on 6-/7-round ChiLow-40. This work assesses the concrete security margins of the ChiLow family.

[1] https://www.iacr.org/testoftime/

Cipher	Setting	#R	Data	Time	Method	Source
Both	ST	3/8	4 KP	2^{126}	MITM	[7]
	ST	4/8	4 KP	2^{102}	MITM	[7]
ChiLow-32	MT	5/8	$2^{23.58}$	2^{32}	Cond.cube	[24]
	MT	5/8	$2^{18.58}$	2^{32}	Bord.cube	[24]
	MT	6/8	2^{40}	2^{120}	Cond.cube	[24]
	MT	6/8	$2^{33.58}$	2^{34}	Bord.cube	[24]
	ST	7/8	2^{8}	$2^{127.78}$	Int	[24]
	ST	8/8	97 KP	$2^{122.6}$	MITM	This paper
	ST	8/8	196 KP	$2^{108.6}$	MITM	This paper
ChiLow-40	ST	6/8	164 KP	$2^{88.9}$	MITM	This paper
	ST	7/8	164 KP	$2^{127.4}$	MITM	This paper

Table 1. Summary of key recovery attacks on ChiLow. The "Setting" column lists the attack setting, including single-tweak (ST) and multi-tweak (MT). Complexities are given in terms of Data (D) and Time (T), where "KP" is short for known plaintext-ciphertext pair. In the "Method" column, "MITM" is short for meet-in-the-middle, "Con" for conditional cube, "Bord" for borderline cube, and "Int" for integral attacks.

2 Specification of ChiLow

ChiLow [7] is a family of tweakable block ciphers and a related PRF designed for embedded code encryption. The main variants, ChiLow-32 and ChiLow-40, adopt a standard substitution–permutation network (SPN) structure.

- ChiLow-32 operates on 32-bit blocks with a 128-bit key and a 64-bit tweak. Its primitive D_{32} decrypts 32-bit instructions. A derived function $D'_{32}(K, T, P)$ produces authentication tags by truncating to $\tau \in \{8, 16\}$ bits.
- ChiLow-40 works on 40-bit blocks with the same key and tweak sizes, using a primitive D_{40} that forces the last 8 bits of the output to zero.

The construction maintains three interacting states:

- a 128-bit *key state* updated by *key schedule*,
- a 64-bit *tweak state* updated by *tweak schedule*,
- a *cipher state* of 32 or 40 bits, updated by *round function*.

Nonlinear layer $\boldsymbol{X}$. The nonlinear layer is built from the χ transformation: for n-bit input $x = (x_0, \ldots, x_{n-1})$,

$$\chi_n(x)_i = x_i + (x_{i+1} + 1)x_{i+2}, \qquad i \in \mathbb{Z}_n.$$

The intertwined function X_n is defined for $n = 2m$ with m even:

$$X_n(x) = \begin{pmatrix} \chi_{m-1}(x[0], \ldots, x[m-1]) \\ \chi_{m+1}(x[m], \ldots, x[2m-1]) \end{pmatrix} + \lambda(x),$$

where $\lambda(x)$ is a linear correction on selected positions:

$$\lambda_i(x) = \begin{cases} x_m + x_{m-3}, & i = m-3, \\ x_{m-1} + x_{m-2}, & i = m-2, \\ x_{m-3} + x_m + x_{m-1}, & i = m-1, \\ x_m + x_{m-2}, & i = m, \\ 0, & \text{otherwise.} \end{cases}$$

Linear layer L. For $x \in \mathbb{F}_2^n$, define

$$(L_n(x))_i = x_{\alpha i+\beta_0} + x_{\alpha i+\beta_1} + x_{\alpha i+\beta_2},$$

with indices modulo n. The parameters are:

- L_{32}: $\alpha = 11$, $\beta_0 = 5$, $\beta_1 = 9$, $\beta_2 = 12$
- L'_{32}: $\alpha = 11$, $\beta_0 = 1$, $\beta_1 = 26$, $\beta_2 = 30$
- L_{40}: $\alpha = 17$, $\beta_0 = 1$, $\beta_1 = 9$, $\beta_2 = 30$
- L_{64}: $\alpha = 3$, $\beta_0 = 1$, $\beta_1 = 26$, $\beta_2 = 50$
- L_{128}: $\alpha = 17$, $\beta_0 = 7$, $\beta_1 = 11$, $\beta_2 = 14$

Key schedule. Initialize K^0 with the 128-bit master key. Each round updates:

$$K^{i+1} = L_{128}\big(X_{128}(K^i \oplus c)\big).$$

Tweak schedule. Initialize $T^0 = T \oplus K[0:63]$. Each round updates:

$$T^{i+1} = L_{64}\big(X_{64}(T^i)\big) \oplus K^{i+1}[0:63].$$

For the last round of the tweak schedule, the description in [7] has a confusing mismatch with the rest of the paper. In [7, Sect.3.2], the formula reflects that the last round tweak update is pure linear: $T^r = L_{64}(T^{r-1})$. However, in both [7, Fig.3 and Fig.4] and the algorithmic description provided in the extended version of the paper [8, Algorithm 1], the last round is mixed with the key state: $T^r = L_{64}(T^{r-1} \oplus K^{r-1})$.

A pure linear mapping at the last round tweak schedule following [7, Sect.3.2] will embed a weakness for ChiLow-40, as guessing 64 bits of the full T^{r-1} is sufficient to get 80 bits round tweakeys TK^{r-1} and TK^r, which leads to an 8-round attack with complexity $2^{127.9}$.

In the following of this paper, we will view the last round of tweak schedule as a linear mixing of tweak state and partial key state:

$$T^r = L_{64}(T^{r-1} \oplus K^{r-1}[0:63]).$$

Round function. Round function follows the SPN structure

$$S^{i+1} = L_n(X_n(S^i)) \oplus TK^{i+1}, \quad 0 \le i \le r-2,$$

with $TK^i = T^i[0:n-1]$, $S^0 = P \oplus TK^0$ and $C = X_n(S^{r-1}) \oplus TK^r$, where $n \in \{32, 40\}$ and r is the number of rounds. We denote the intermediate states after the X and L layers as

$$S^i \xrightarrow{X} A^i \xrightarrow{L} B^i \xrightarrow{ATK} S^{i+1}.$$

Security claims. Both ChiLow-32 and ChiLow-40 claims 128-bit security with 8 rounds and the following constraints:

- Queries under the same tweak must not exceed 2^8.
- Total queries must remain below 2^{40}.

3 Cryptanalytic Observations of ChiLow

3.1 Full Diffusion Rounds.

This might be a common pitfall for ultra-low latency designs with a small state but large tweakey size. In the setting of ChiLow-32, the tweak is 64 bits and key is 128 bits. As the state is only 32 bits, a total of $(64 + 128)/32 = 6$ rounds will be needed for full tweakey diffusion. A trivial thought exploiting this observation is to guess 3 tweakeys in the forward direction and 3 tweakeys in the backward direction, which constitutes a trivial 6-round MITM attack.

3.2 Diffusion of XL, XLX, and XLXL.

We will discuss the diffusion for components of the non-linear and linear layer of ChiLow, denoted repectively by X and L. The novel ChiChi layer X has a very clean structure, it is trivial to see that given 1 output bit of X, its value is dependent to 3 input bits. For linear layer of L, same observation applies, excepts the relations are linear.

We next look at one round of ChiLow begins with ATK, abbreviated by XL, namely $L(X(S \oplus TK))$, where S denote the input state and TK is the round tweakey. For 1 output bit of XL, its value depend on 9 input key bits, in which 3 bits are linearly involved and the other 6 are nonlinearly involved.

We further look at XLX, denoting two iterations of XL excluding the last linear layer. We are interested in how many tweakey bits that are nonlinearly involved for an output bit of XLX. We compute the algebraic formulation of XLX operating on a 32-bit state and on a 40-bit state with the SAGEMATH tool , namely XLX_{32} and XLX_{40}. However, from our experiment, which is reflected as Table 2 and Table 3, we can see that 1 output bit of XLX_{32} corresponds to a range from 20 to 26 bits of nonlinear key bits, and XLX_{40} a range from 22 to 25 nonlinear key bits.

We finally look at XLXL, denoting 2 round of XL. Performing the similar experiment on XLX, we obtain the count for nonlinear key bits for 1 bit output respectively for XLXL_{32} and XLXL_{40}, reflected in Table 4 and Table 5.

Dependencies	Output bit indices
20	15, 16
22	14
23	13, 30, 31
24	4, 26, 27
25	0, 1, 2, 3, 5, 6, 7, 8, 10, 11, 17, 19, 20, 23, 25, 28, 29
26	9, 12, 18, 21, 22, 24

Table 2. Count of non-linear key bits involved for 1 output bit of XLX_{32}

Dependencies	Output bit indices
22	19, 20, 38, 39
23	6, 9, 10, 17, 29, 36
24	0, 1, 2, 3, 4, 5, 7, 8, 11, 12, 13, 15, 16, 18, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35, 37
25	14, 21, 22, 23

Table 3. Count of non-linear key bits involved for 1 output bit of XLX_{40}

Dependencies	Output bit indices
37	14, 24, 31
38	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30

Table 4. Count of non-linear key bits involved for 1 output bit of XLXL_{32}

Dependencies	Output bit indices
41	11, 23, 32, 39
42	0, 5, 7, 8, 9, 15, 16, 17, 18, 20, 22, 25, 27, 29, 30, 31, 36, 37, 38
43	1, 2, 3, 6, 10, 12, 13, 19, 24, 33, 34
44	4, 14, 21, 28, 35
45	26

Table 5. Count of non-linear key bits involved for 1 output bit of XLXL_{40}

3.3 Linear Structures of X.

The nonlinear layer of Chichi X with even dimension $2m$ is designed from concatenating χ_{m-1} and χ_{m+1} followed by a linear mapping that mixes the the two small χs. The structural resemblance between X and χ enables transferable observations from previous attacks on Keccak, namely, the linear structure technique proposed by Guo, Liu and Song at Asiacrypt 2016 [18]. Let a_i and b_i to denote the input bits and output bits of X. Suppose we know the first 2 output

bits of X, namely b_0, b_1, similar to [18], we are able to construct an equation between a and b:

$$b_0 = a_0 + (b_1 + 1) \cdot a_2$$

Furthermore, when $b_1 = 1$, we have $b_0 = a_0$. This observation leads to a conditional matching technique that leads to more efficient attacks but with more known (P, C) pairs, illustrated in Section 5.2.

4 Relevant Techniques in MITM attacks

In this section, we first introduce the Meet-in-the-Middle key recovery attack structure. We then review three three relevant techniques, namely matching amplification, key cancellation, and conditional matching. We are not intended to claim any originality for those techniques. Finally, we propose an integrated attack framework, which is later applied to the attacks on ChiLow.

Table 6. Common notations for three-subset MITM attacks

K_f, K_b	Three independent subspaces of the key space.
F_{K_f}	Forward chunk under (K_f, K_2).
G_{K_b}	Backward chunk under (K_b, K_2).
$M_{K_f \| K_b}$	Middle chunk, where partial-match is performed.
k_f, k_b	Guessed key values for K_f, K_b
m	Number of partial-match bits used in the MITM stage.
t	Number of plaintext/ciphertext pairs used in the key testing stage.

4.1 The Meet-in-the-Middle Attack Structure

The MITM attack consists of two stages: the MITM stage and the Key Testing stage. This technique has since been applied to many block ciphers and hash-based designs [22,30,21,9,27,15,6]. With notations listed in Table 6, an overview of this method is shown below:

Chunk division The block cipher E_K is divided into three chunks as

$$E_K(P) = G_{K_b}\big(M_{K_f \| K_b}\big(F_{K_f}(P)\big)\big) = C,$$

where the key space is partitioned into independent subspaces K_f,K_2, and K_b.

MITM stage The chunks now have the following structure:

$$P \xrightarrow{F_{K_f}} S_f \xleftrightarrow{M_{K_f \| K_b}} S_b \xleftarrow{G^{-1}_{K_b}} C.$$

One can independently compute F and G to intermediate states S_f and S_b, and perform an m-bit partial match with M. This reduces the search space from $2^{|K_f|+|K_b|}$ to $2^{|K_f|+|K_b|-m}$, with a time complexity of $2^{|K_f|} + 2^{|K_b|}$.

Key testing stage The surviving key candidates from the MITM stage are tested using some plaintext/ciphertext pairs. This step ensures identification of the correct key, while its cost is usually negligible compared to the MITM stage.

The overall time complexity can be evaluated as

$$2^{|K_f|} + 2^{|K_b|} + 2^{|K_f|+|K_b|-m}. \tag{1}$$

4.2 Matching Amplification with Multiple Known Pairs

In MITM key recovery attacks, if the subspaces K_f and K_b are chosen independent from the value of (P, C), we can use multiple known pairs (MulKP) to amplify the degrees of matching m at the MITM stage.

We denote the number of pairs as t. For forward and backward key guesses k_f and k_b, we can compute F and G independently for t pairs of known (P, C). Then, (k_f, k_b) proceeds to the key testing stage if and only if it passed the m-bit filter for all t pairs. The filtering power is thus amplified to $t \cdot m$.

4.3 Key-cancellation Technique

Building upon the MulKP MITM structure, we observe that by computing the difference between two pairs, we can cancel the round keys if the round keys are linearly involved (i.e. before non-linear layer).

Consider the following toy example of an SPN cipher, suppose we have S and Z computed with forward and backward guessed key bits, and we will need to propagate through a linear layer L followed by ATK.

$$S \xrightarrow{L \circ ATK^i} W \xleftrightarrow{Match} Y \xleftarrow{L^{-1} \circ ATK^j} Z$$

A trivial idea would be guessing further the involved key bits in TK^i and TK^j, such that we can have a matching in between. This approach will lead to more guessed key bits, verifying them will lead to an increase of attack complexity.

However, we can use a pair of known (P, C) pairs and match on their differences to cancel the effect of TK^i and TK^j. We assume we have (P_0, C_0) and (P_1, C_1), with corresponding computed states (S_0, Z_0) and (S_1, Z_1). In the forward direction, we have

$$W_0 = L(S_0 \oplus TK^i)$$
$$W_1 = L(S_1 \oplus TK^i)$$

The difference between T_0 and T_1 is independent from TK^i:

$$\Delta W = L(S_0 \oplus TK^i) \oplus L(S_1 \oplus TK^i) = L(S_0 \oplus S_1) = L(\Delta S).$$

The same logic works for the backward direction. Thus, we can perform matching between $L(\Delta S)$ and $L(\Delta Z)$.

4.4 Conditional Matching

There are cases where the matching filter happens with a condition on the value of forward or backward chunk, namely conditional matching. Consider the same toy example, but now suppose we already propagate S and Z respectively to W and Y:

$$S \rightarrow W \xleftrightarrow{Match^{cond}} Y \leftarrow Z$$

Suppose the condition is on the value of Y with probability p.

In this scenario, we can use more known pairs to compensate p and construct conditional matching. We simply prepare more known pairs, for each backward key guess, we collect the pairs that lead to backward chunk values satisfying this condition into a set Φ. In the forward direction, we only use the pairs in Φ to compute forward matching values. The resulting attack is still deterministic.

4.5 Integrated MITM Key Recovery Attack Framework

The following algorithm reflects the integration of the above techniques into MITM key recovery attacks. We use multiple pairs to amplify the matching filter, use difference between pairs to cancel linearly involved round tweakeys, and match with a probability p conditioned on the backward chunk value.

Algorithm 1: A multi-pair MITM attack with key cancellation

```
for k_b ∈ 2^{|K_b|} do
    for i ∈ {0, 1, ..., (t+1)·1/p} do
        Compute G_{K_b}(C_i);
        if condition is met then
            Φ ← i;
            obtain value for matching α_i;
    T[β_{Φ[0]} ⊕ β_{Φ[1]}, ..., β_{Φ[0]} ⊕ β_{Φ[t]}] ← k_b
for k_f ∈ 2^{|K_f|} do
    for i ∈ Φ do
        Compute F_{K_f}(P_i), obtain value for matching α_i;
    for entries in T[α_{Φ[0]} ⊕ α_{Φ[1]}, ..., α_{Φ[0]} ⊕ α_{Φ[t]}] do
        Check for a full match;
```

To get $t \cdot m$ bits of filtering, we will need $t+1$ pairs in Φ to construct t differences. As the conditional matching occurs with probability p, we will need $(t+1)/p$ known pairs under the same secret key (and tweak). The final time complexity becomes:

$$(t+1) \cdot \left(2^{|K_b|} \cdot \frac{1}{p} + 2^{|K_f|}\right) + 2^{|K_f|+|K_b|-t \cdot m}. \tag{2}$$

5 MITM key recovery attacks on 8-round ChiLow-32

We propose a known plaintext attack on full-round ChiLow with low data and single secret tweakey using the MITM framework.

5.1 An 8-round Attack with 2-round Matching

For a known (P, C) pair, we can guess the full TK_0, TK_1, TK_2 in the forward direction and the full TK_6, TK_7, TK_8 in the backward direction. Thus, we can obtain two full states S^2 and S^6 respectively from the forward and backward direction. In the forward direction, we can further propagate S^2 to B^2, as there is no tweakey involved. Hence, we are left with the following layers in between:

$$B^2 \xrightarrow{ATK^3} S^3 \xrightarrow{X} A^3 \xrightarrow{L} B^3 \xrightarrow{ATK^4} S^4 \xrightarrow{X} A^4 \xrightarrow{L} B^4 \xrightarrow{ATK^5} S^5.$$

In the backward direction, as the TK^5 is linearly involved from S^5 to A^4, we can apply the key cancellation technique following Section 4.3 and have $\Delta A^4 = L(\Delta S^5)$. Thus, we bypass the influence of TK^5 in the backward direction and are left with the following structure:

$$\Delta B^2 \xrightarrow{ATK^3, X, L, ATK^4, X} \Delta A^4 \xleftrightarrow{Match} L(\Delta S^5).$$

Noticed that the transition from B^2 to A^4 undergoes a XLX structure. From Table 2, we note that the minimal number of key dependencies to compute 1 bit of ΔA^4 is 20. Thus, by guessing 20 bits in TK^3 and TK^4, we are able to compute 1 bit of ΔA^4. Without loss of generality, we choose 1-bit matching at $\Delta A^4[16]$. The key bits we need to guess are

$$TK^3[0, 1, 2, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20],\ TK^4[17, 18].$$

Following Equation (2), we have $|K_b| = 96$, $|K_f| = 96 + 20$. The overall complexity is:

$$(t+1) \cdot (2^{96+1} + 2^{96+20}) + 2^{96+20+96-t}$$

Using $m = 1$ and $t = 96$, we have the overall complexity as $97 \cdot 2^{116} \approx 2^{122.6}$, with 97 known pairs.

5.2 More Efficient Attack with Conditional Matching

A quick improvement upon the attack described above stems from the linear structure technique proposed by Guo, Liu and Song at Asiacrypt 2016 [18]. Linear structures provide a nice way to perform matching around X, preventing from penetrating full XLX from the forward direction. We thus have the following structure.

$$\Delta B^2 \xrightarrow{ATK^3,X,L,ATK^4} \Delta S^4 \xleftrightarrow{Match^X} \Delta A^4 = L(\Delta S^5).$$

We choose the matching between $\Delta S^4[0]$ and $L(\Delta S^5[0])$. In the backward direction, to linearize the X layer at position 0, we need to condition $A^4[1] = 1$. As $A^4[1] = L(S^5 \oplus TK^5)[1]$ and S^5 is known, we will need to guess 1-bit value of $L(TK^5)[1]$. In the forward direction, from ΔB^2 we need to guess the 6 nonlinear bits of TK^3 to compute ΔS^4, as the linear key bits are cancelled.

Following Equation (2), we have $|K_b| = 96 + 1$, $|K_f| = 96 + 6$. The overall complexity is:

$$(t+1) \cdot (2^{96+1} \cdot 2 + 2^{96+6}) + 2^{96+6+96+1-t}$$

Using $m = 1$ and $t = 97$, we have the overall complexity approximately as $98 \cdot 2^{102} \approx 2^{108.6}$, with $(97 + 1) \cdot 2 = 196$ known pairs.

6 MITM key recovery attacks on 8-round ChiLow-40

The 2-round matching technique used in ChiLow-32 is not sufficient in ChiLow-40, as the complexity will be dominated by the guess of three full forward round keys TK^0, TK^1, TK^2, which should be around 120. The overall complexity will likely to be too close to the generic bound. In the attacks on ChiLow-40, we combine the two techniques used in ChiLow-32, and propose a 3-round matching.

6.1 A 6-round Attack with 3-round Matching

For a known (P, C) pair, we guess TK_0 in the forward direction and TK^5, TK^6 in the backward direction. Thus, we can obtain two full states S^0 and S^4 respectively from the forward and backward direction. Similar to Section 5.2, we apply the key-cancellation technique and match with differences. However, in this attack, the matching is 3-round matching starting from B^1:

$$\Delta B^0 \xrightarrow{ATK^1,X,L,ATK^2,X,L,ATK^3} \Delta S^3 \xleftrightarrow{Match^X} \Delta A^3 = L(\Delta S^4).$$

In the forward direction, note that the transition from B^0 to S^3 undergoes a XLXL structure (TK^3 is cancelled by computing differences). From Table 5, to compute 1 bit of S^3 in the forward direction, the minimal nonlinear key bit count is 41. Without loss of generality, we choose $S^3[11]$. Thus, we need to guess additional 41 bit, namely:

$$TK^1[1-12,\ 14-24,\ 27-33,\ 35-39], \quad TK^2[0, 18, 29, 30, 37, 38]$$

In the backward direction, similar to Section 5.2, we pose 1 bit condition on $A^3[12]$, with additional 1-bit guessing of $L(TK^4)[12]$.

Following Equation (2), we have $|K_b| = 80 + 1$, $|K_f| = 40 + 41$. The overall complexity is:

$$(t+1) \cdot (2^{80+1} \cdot 2 + 2^{40+41}) + 2^{40+41+80+1-t}$$

Using $m = 1$ and $t = 81$, we have the overall complexity approximately as $82 \cdot 2^{82.5} \approx 2^{88.9}$, with $(81+1) \cdot 2 = 164$ known pairs.

6.2 A natural 7-round Attack

A natural 7-round attack can be obtained by guessing one more round tweakey in the forward direction. The overall complexity is:

$$(t+1) \cdot (2^{80+1} \cdot 2 + 2^{40+40+41}) + 2^{40+40+41+80+1-t}$$

Using $m = 1$ and $t = 81$, we have the overall complexity approximately as $82 \cdot 2^{121} \approx 2^{127.4}$, with $(81+1) \cdot 2 = 164$ known pairs.

References

1. Aoki, K., Guo, J., Matusiewicz, K., Sasaki, Y., Wang, L.: Preimages for Step-Reduced SHA-2. In: Matsui, M. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 5912, pp. 578–597. Springer (2009), https://doi.org/10.1007/978-3-642-10366-7_34
2. Aoki, K., Sasaki, Y.: Preimage Attacks on One-Block MD4, 63-Step MD5 and More. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC. Lecture Notes in Computer Science, vol. 5381, pp. 103–119. Springer (2008), https://doi.org/10.1007/978-3-642-04159-4_7
3. Aoki, K., Sasaki, Y.: Meet-in-the-Middle Preimage Attacks Against Reduced SHA-0 and SHA-1. In: Halevi, S. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 5677, pp. 70–89. Springer (2009), https://doi.org/10.1007/978-3-642-03356-8_5
4. Bao, Z., Ding, L., Guo, J., Wang, H., Zhang, W.: Improved meet-in-the-middle preimage attacks against AES hashing modes. IACR Trans. Symmetric Cryptol. **2019**(4), 318–347 (2019), https://doi.org/10.13154/tosc.v2019.i4.318-347
5. Bao, Z., Dong, X., Guo, J., Li, Z., Shi, D., Sun, S., Wang, X.: Automatic Search of Meet-in-the-Middle Preimage Attacks on AES-like Hashing. In: EUROCRYPT I. pp. 771–804 (2021), https://doi.org/10.1007/978-3-030-77870-5_27
6. Bao, Z., Guo, J., Shi, D., Tu, Y.: Superposition Meet-in-the-Middle Attacks: Updates on Fundamental Security of AES-like Hashing. In: CRYPTO I. pp. 64–93 (2022), https://doi.org/10.1007/978-3-031-15802-5_3
7. Belkheyar, Y., Derbez, P., Ghosh, S., Leander, G., Mella, S., Perrin, L., Rasoolzadeh, S., Stennes, L., Sun, S., Assche, G.V., Vizár, D.: Chilow and chichi: New constructions for code encryption. In: EUROCRYPT (1). Lecture Notes in Computer Science, vol. 15601, pp. 212–243. Springer (2025)
8. Belkheyar, Y., Derbez, P., Ghosh, S., Leander, G., Mella, S., Perrin, L., Rasoolzadeh, S., Stennes, L., Sun, S., Assche, G.V., Vizár, D.: ChiLow and ChiChi: New constructions for code encryption. Cryptology ePrint Archive, Paper 2025/308 (2025), https://eprint.iacr.org/2025/308
9. Biham, E., Dunkelman, O., Keller, N., Shamir, A.: New attacks on IDEA with at least 6 rounds. J. Cryptol. **28**(2), 209–239 (2015), https://doi.org/10.1007/s00145-013-9162-9
10. Bogdanov, A., Rechberger, C.: A 3-Subset Meet-in-the-Middle Attack: Cryptanalysis of the Lightweight Block Cipher KTANTAN. In: SAC. pp. 229–240 (2010), https://doi.org/10.1007/978-3-642-19574-7_16
11. Canteaut, A., Naya-Plasencia, M., Vayssière, B.: Sieve-in-the-Middle: Improved MITM Attacks. In: Canetti, R., Garay, J.A. (eds.) CRYPTO I. Lecture Notes in Computer Science, vol. 8042, pp. 222–240. Springer (2013), https://doi.org/10.1007/978-3-642-40041-4_13
12. Chen, S., Guo, J., List, E., Shi, D., Zhang, T.: Diving Deep into the Preimage Security of AES-Like Hashing. In: Joye, M., Leander, G. (eds.) EUROCRYPT I. Lecture Notes in Computer Science, vol. 14651, pp. 398–426. Springer (2024), https://doi.org/10.1007/978-3-031-58716-0_14
13. Chen, S., Guo, J., List, E., Shi, D., Zhang, T.: Scrutinizing the security of AES-based hashing and one-way functions. Cryptology ePrint Archive, Paper 2025/792 (2025), https://eprint.iacr.org/2025/792
14. Diffie, W., Hellman, M.E.: Special feature exhaustive cryptanalysis of the NBS data encryption standard. IEEE Computer **10**(6), 74–84 (1977), https://doi.org/10.1109/C-M.1977.217750

15. Dong, X., Hua, J., Sun, S., Li, Z., Wang, X., Hu, L.: Meet-in-the-Middle Attacks Revisited: Key-Recovery, Collision, and Preimage Attacks. In: CRYPTO III. pp. 278–308 (2021), https://doi.org/10.1007/978-3-030-84252-9_10
16. Dong, X., Zhao, B., Qin, L., Hou, Q., Zhang, S., Wang, X.: Generic mitm attack frameworks on sponge constructions. In: Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part IV. pp. 3–37 (2024), https://doi.org/10.1007/978-3-031-68385-5_1
17. Guo, J., Ling, S., Rechberger, C., Wang, H.: Advanced Meet-in-the-Middle Preimage Attacks: First Results on Full Tiger, and Improved Results on MD4 and SHA-2. In: ASIACRYPT. pp. 56–75 (2010), https://doi.org/10.1007/978-3-642-17373-8_4
18. Guo, J., Liu, M., Song, L.: Linear structures: Applications to cryptanalysis of round-reduced keccak. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 249–274 (2016). https://doi.org/10.1007/978-3-662-53887-6_9, https://doi.org/10.1007/978-3-662-53887-6_9
19. Hou, Q., Dong, X., Qin, L., Zhang, G., Wang, X.: Automated Meet-in-the-Middle Attack Goes to Feistel. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT III. Lecture Notes in Computer Science, vol. 14440, pp. 370–404. Springer (2023), https://doi.org/10.1007/978-981-99-8727-6_13
20. Hua, J., Dong, X., Sun, S., Zhang, Z., Hu, L., Wang, X.: Improved MITM cryptanalysis on streebog. IACR Trans. Symmetric Cryptol. **2022**(2), 63–91 (2022), https://doi.org/10.46586/tosc.v2022.i2.63-91
21. Isobe, T.: A single-key attack on the full GOST block cipher. J. Cryptol. **26**(1), 172–189 (2013), https://doi.org/10.1007/s00145-012-9118-5
22. Isobe, T., Shibutani, K.: Security analysis of the lightweight block ciphers xtea, LED and piccolo. In: Information Security and Privacy - 17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, July 9-11, 2012. Proceedings. pp. 71–86 (2012), https://doi.org/10.1007/978-3-642-31448-3_6
23. Khovratovich, D., Rechberger, C., Savelieva, A.: Bicliques for Preimages: Attacks on Skein-512 and the SHA-2 Family. In: Canteaut, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 7549, pp. 244–263. Springer (2012), https://doi.org/10.1007/978-3-642-34047-5_15
24. Peng, S., He, J., Hu, K., Niu, Z., Rasoolzadeh, S., Wang, M.: Cryptanalysis of ChiLow with cube-like attacks. Cryptology ePrint Archive, Paper 2025/1581 (2025), https://eprint.iacr.org/2025/1581
25. Qin, L., Hua, J., Dong, X., Yan, H., Wang, X.: Meet-in-the-Middle Preimage Attacks on Sponge-Based Hashing. In: EUROCRYPT IV. pp. 158–188 (2023), https://doi.org/10.1007/978-3-031-30634-1_6
26. Sasaki, Y.: Meet-in-the-Middle Preimage Attacks on AES Hashing Modes and an Application to Whirlpool. In: Joux, A. (ed.) FSE. Lecture Notes in Computer Science, vol. 6733, pp. 378–396. Springer (2011), https://doi.org/10.1007/978-3-642-21702-9_22
27. Sasaki, Y.: Integer linear programming for three-subset meet-in-the-middle attacks: Application to GIFT. In: Advances in Information and Computer Security - 13th International Workshop on Security, IWSEC 2018, Sendai, Japan, September 3-5, 2018, Proceedings. pp. 227–243 (2018), https://doi.org/10.1007/978-3-319-97916-8_15

28. Sasaki, Y., Aoki, K.: Preimage Attacks on 3, 4, and 5-Pass HAVAL. In: Pieprzyk, J. (ed.) ASIACRYPT. Lecture Notes in Computer Science, vol. 5350, pp. 253–271. Springer (2008), https://doi.org/10.1007/978-3-540-89255-7_16
29. Sasaki, Y., Aoki, K.: Finding Preimages in Full MD5 Faster Than Exhaustive Search. In: Joux, A. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 5479, pp. 134–152. Springer (2009), https://doi.org/10.1007/978-3-642-01001-9_8
30. Sasaki, Y., Wang, L., Sakai, Y., Sakiyama, K., Ohta, K.: Three-subset meet-in-the-middle attack on reduced XTEA. In: Progress in Cryptology - AFRICACRYPT 2012 - 5th International Conference on Cryptology in Africa, Ifrance, Morocco, July 10-12, 2012. Proceedings. pp. 138–154 (2012), https://doi.org/10.1007/978-3-642-31410-0_9
31. Schrottenloher, A., Stevens, M.: Simplified MITM Modeling for Permutations: New (Quantum) Attacks. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO III. Lecture Notes in Computer Science, vol. 13509, pp. 717–747. Springer (2022), https://doi.org/10.1007/978-3-031-15982-4_24
32. Schrottenloher, A., Stevens, M.: Simplified Modeling of MITM Attacks for Block Ciphers: New (Quantum) Attacks. IACR Trans. Symmetric Cryptol. **2023**(3), 146–183 (2023), https://doi.org/10.46586/tosc.v2023.i3.146-183