Controlled Differentials in Key-Then-Hash Functions: No Absolute Weak-Key Classes and Key Recovery through MACs

Jonathan Fuchs

Radboud University, Nijmegen, The Netherlands
jonathan.fuchs@ru.nl

Abstract. In their CRYPTO 2023 paper, Fuchs, Rotella, and Daemen reduced the universality of key-then-hash (KTH) functions built from public permutations to differential properties of the underlying permutation. Subsequent work analyzed KTH Parallel with fixed-length public functions and related variable-length universality bounds [12,13]. We develop a corresponding framework for key-recovery attacks on KTH-based MACs.

We first revisit the weak-key classes identified by Handschuh and Preneel for NH, NMH*, WH, and Square Hash. For the KTH attacks considered here, these classes are solution sets of controlled differentials. These translated sets may still satisfy HP08's broad operational weak-key criterion. Our claim is different: KTH offset invariance translates every such solution set through every absolute key with unchanged probability and oracle-query cost, so the attacks do not identify intrinsically exceptional absolute keys. For these controlled attacks, this gives a strong KTH form of the UHF "regularity" sought by Handschuh and Preneel: the complete attack profile is homogeneous over absolute key space.

We then show that, for every fixed-length KTH map with no probability-one nonzero differential, any two distinct key-prefix candidates can be distinguished by a generalized differential test. If its generalized derivatives are affine, one retained successful fixed-output equal-length differential leaves an affine coset and at most $\lfloor \log_2 \#S_0 \rfloor$ further suitably chosen successful events determine the prefix, under the stated computational and interface assumptions.

Finally, we apply the framework to NH and Xoodoo[3]. Under reusable-reference WC(S), unequal-length NH image tests simultaneously test one candidate for each word of a multiplication pair and recover both w-bit words with one generation query and at most $2^w + 1$ verification queries, approximately half the verification-query cost of the Handschuh–Preneel recovery. For Serial[Xoodoo[3]], published three-round trails yield recovery of a reusable 384-bit KTH key block in 2^{42} expected differential trials, or 2^{43} MAC-oracle calls.

Keywords: Message authentication codes · weak-key classes · key recovery · key-then-hash · differential cryptanalysis · Xoodoo

1 Introduction

Universal hash functions (UHFs) underpin protected-hash and Wegman–Carter-family MACs, where universality bounds the collision or fixed-output-difference event needed for forgery [24–26, 28]. Fuchs, Rotella, and Daemen [11] formalized key-then-hash (KTH) functions and showed that for Serial and Parallel built from public permutations, universality reduces to differential properties of the permutation. Ghosh et al. [13] analyzed KTH Parallel constructions using fixed-length public functions. In this public-function setting, differential and image probabilities govern the variable-length bounds used below [12]. We develop the corresponding viewpoint for key recovery through KTH-based MACs.

The motivation is the classical analysis of Handschuh and Preneel [15]. They interpret several key-reuse attacks as evidence of weak-key classes, namely efficiently detectable subsets with unexpectedly favorable attack behavior. Their security discussion makes this interpretation operational. They note that parties could check for weak keys during key generation, although avoiding all such combinations can substantially increase its cost [15, p. 145]. Their conclusion describes many UHF-based MACs as "very brittle" because of simple combinatorial properties, recommends avoiding key reuse, and contrasts them with a more "robust" AES-based alternative [15, pp. 157–158]. They then ask for a formal notion of UHF "regularity" that could alleviate concerns related to weak keys [15, p. 158]. Thus the weak-key label is used to distinguish key values whose absolute placement is relevant to robustness. Such screening can remove the phenomenon only if some absolute key values are genuinely more vulnerable than the remainder. We do not argue that translated solution sets fail HP08's broad criterion. The question is whether, for KTH, that criterion distinguishes intrinsically exceptional absolute keys.

For NH, the tension is already visible in their own attacks. They single out equal subkeys, while their stronger recovery tests arbitrary relations $k_2 - k_1 = d$ [15, pp. 151–156]. As d ranges over $\mathbb{Z}/2^w\mathbb{Z}$, these relation classes partition the two-word key space into equal parts. The case $d = 0$ is therefore not cryptographically privileged. The arbitrary-d attack is thus evidence, already inside HP08, that equality itself is not the cryptanalytic invariant. This is not a disagreement with the attack or with its complexity. It is a disagreement about what the attack witnesses: a special absolute key value, or a reusable algebraic relation that can be moved through key space. Appendix A gives the concrete derivation and the analogous KTH instantiations.

KTH explains this systematically. It satisfies

$$F_{\mathbf{k}+\mathbf{o}}(\mathbf{m}) = F_{\mathbf{k}}(\mathbf{m} + \mathbf{o}),$$

so a message translation moves a controlled collision or fixed-difference solution set through absolute key space. The translation preserves class cardinality, uniform-key success probability, message lengths, and oracle-query cost. Theorem 1 strengthens this observation to the complete controlled-test landscape. Every absolute key has an isomorphic Handschuh–Preneel attack profile for the

attacks in our scope, so these attacks cannot distinguish a nontrivial absolute weak-key class. This changes the interpretation, not the attacks. The membership tests still identify differential solution sets and still support key recovery, but the absolute placement of those sets carries no cryptographic privilege. For these controlled KTH attacks, this gives an unusually strong form of the regularity sought in HP08: the complete attack landscape is translation-isomorphic at every absolute key. Accordingly, future KTH design should treat the differential and image probabilities underlying the applicable universality bounds and the permitted hash-key reuse as the relevant pre-success criteria. Screening particular absolute key values adds no protection against this mechanism.

The solution-set view also exposes a better NH attack. Handschuh and Preneel first determine the additive relation between the two subkeys of an NH multiplication pair and then determine one absolute word value. An unequal-length KTH image test instead exploits the zero preimage of multiplication and simultaneously tests one value of each subkey. Under WC(S), the original reference tag itself tests whether the appended branch contributes zero. Corollary 6 recovers both w-bit words with at most $2^w + 1$ verification queries, about half the verification-query cost of their recovery.

We then prove the generic key-recovery counterpart of differential universality. For a fixed length ℓ, let $F_\ell : \mathrm{G}^\ell \to \mathrm{G}'$ be the public KTH map. If no nonzero equal-length generalized differential has probability one, every pair of distinct key-prefix candidates is separated by some differential test. If its generalized derivatives are affine, suitably chosen directions make at most $\lfloor \log_2 \#S_0 \rfloor$ additional successful differential events sufficient after one retained success, under the stated computational assumptions. Corollary 4 separately accounts for the raw trials needed to obtain those successes. Applying the framework to published three-round Xoodoo trails gives recovery of a reusable 384-bit Serial KTH key block in 2^{42} expected differential trials, or 2^{43} MAC-oracle calls.

Forgery-to-key-recovery is not itself new. Minematsu and Tsunoo connect differential probability to provably secure MAC design [21]. Section 1.2 distinguishes our results from prior weak-key, polynomial-WCS, and partition-oracle attacks.

1.1 Contributions

- **Structural result.** The complete controlled HP08-style KTH attack landscape is translation-isomorphic at every absolute key, giving a strong KTH form of HP08's regularity objective and ruling out nontrivial absolute weak-key classes.
- **Generic separation.** $\mathsf{MDP}_{F_\ell} < 1$ separates every candidate pair and gives information-theoretic recovery.
- **Affine recovery.** At most $\lfloor \log_2 \#S_0 \rfloor$ selected successful affine equations suffice. Raw trials are separate.
- **Improved NH recovery.** Reusable-reference WC(S) recovers both words with one generation query and at most $2^w + 1$ verification queries, about half the HP08 verification-query cost.

– **Serial[Xoodoo[3]].** A reusable 384-bit block is recovered in 2^{42} expected differential trials, or 2^{43} MAC-oracle calls.

1.2 Scope and Related Work

The weak-key statement is deliberately scoped. We do not claim that weak keys never exist or that every construction in [15] is KTH. Theorem 1 concerns controlled tests on native KTH message domains, and restricted protocols may block translations.

These results do not contradict the universality bounds or security claims established for the KTH constructions of [11–13]. Rather, they characterize the consequences of controlled differential events when a reusable KTH hash key is exposed through the MAC interfaces considered here.

Procter and Cid show that weak-key forgeries in polynomial MACs can be engineered through the root structure of polynomial evaluation [23]. As summarized explicitly by Armour and Cid, these forgeries can effectively test attacker-chosen target key sets, which leads naturally to a partition-oracle interpretation [1]. This exposes how permissive the weak-key notion can become for polynomial hashing. Our conclusion is analogous in spirit but follows from a different structural mechanism. KTH translation equivariance does not rely on engineering roots or arbitrary target sets. Instead it transports every realizable controlled differential solution set through key space, making the complete controlled attack profile homogeneous over absolute keys. Luykx and Preneel give polynomial WCS/GCM forgery and hash-key-recovery attacks [19], and Nandi repairs their quantitative WCS analysis and gives tight hash-key-recovery and forgery attacks [22]. We do not claim weak-key testing, partitioning, or forgery-to-key-recovery as new. Our contributions are the KTH translation-isomorphism and absence of privileged absolute weak-key classes, generic candidate separation under $\mathsf{MDP} < 1$, affine generalized-derivative recovery, and the NH image-recovery attack.

Zheng and Wang [29] derive general UHF weak-key classes from combinatorial properties, while Theorem 1 removes absolute placement from the controlled KTH solution-set interpretation. Wang et al. [27] study related-key almost universality, whereas our unknown key is fixed and only messages are translated. KTH universality is developed in [11, 13], with further public-function KTH analysis in [12] and multi-query WC(S) bounds in [3].

Reduced-round attacks on Xoofff, Xoodoo-AE, and Xoodyak use different targets and interfaces [6, 9, 30, 31]. Our result instead recovers a reusable KTH block through WC(S) or PH from three-round trail data together with exact differential information [5, 20].

2 Notation and Definitions

2.1 Notation

Let $(\mathrm{G}, +)$ and $(\mathrm{G}', +)$ be finite abelian groups. Elements of G are called blocks. The set of nonempty block strings of lengths $1, \ldots, \kappa$ is denoted by $\mathrm{BS}(\mathrm{G}, \kappa)$, as

in [11]. The empty string is not an element of $\mathrm{BS}(\mathrm{G}, \kappa)$. We write a message as $\mathbf{m} = (m_1, \ldots, m_{|\mathbf{m}|})$ and the reusable hash key as $\mathbf{k} = (k_1, \ldots, k_\kappa) \in \mathrm{G}^\kappa$. We use $\#A$ for the cardinality of a finite set A.

A keyed hash family is written

$$F : \mathrm{G}^\kappa \times \mathrm{BS}(\mathrm{G}, \kappa) \to \mathrm{G}',$$

and $F_{\mathbf{k}}(\cdot)$ denotes the function obtained after fixing the key $\mathbf{k}$.

2.2 Universal Hash Functions

We use the standard notions of universality [26]. A keyed hash function F is ε-universal if, for every pair of distinct messages $\mathbf{m}, \mathbf{m}^*$,

$$\Pr_{\mathbf{k}}[F_{\mathbf{k}}(\mathbf{m}) = F_{\mathbf{k}}(\mathbf{m}^*)] \leq \varepsilon,$$

and ε-Δ-universal if, for every such pair and every $\Delta \in \mathrm{G}'$,

$$\Pr_{\mathbf{k}}[F_{\mathbf{k}}(\mathbf{m}) - F_{\mathbf{k}}(\mathbf{m}^*) = \Delta] \leq \varepsilon,$$

where $\mathbf{k}$ is uniform over G^κ.

2.3 Message Authentication Codes

We consider two standard UHF-based MAC settings.

Wegman-Carter(-Shoup) Authenticators. Let g be a secure keyed function with output in G'. A tag has the form

$$T = F_{\mathbf{k}}(\mathbf{m}) + g(k', N),$$

where the key of g is independent of $\mathbf{k}$ [24, 28]. A valid tag for $(\mathbf{m}, N)$ therefore lets the attacker test a known difference $F_{\mathbf{k}}(\mathbf{m}) - F_{\mathbf{k}}(\mathbf{m}^*) = \Delta$ by verifying $T - \Delta$ for $(\mathbf{m}^*, N)$. We count both a *reusable-reference* interface, where a reference remains usable after every verification answer, and a *one-shot-nonce* interface, where each test needs a fresh authenticated reference. These interfaces change generation counts but not the tested UHF relations or candidate-set arguments.

Protected Hash Authenticators. Bellare, Canetti, and Krawczyk describe the hash-then-protect construction in which the message is first hashed and a keyed authentication function is then applied to the hash value [2]. Following the KTH terminology of [11], we refer to this setting as protected hash (PH):

$$\mathbf{m} \longmapsto g(k', F_{\mathbf{k}}(\mathbf{m})).$$

We use a collision-transparent PH model in which equality of protected tags on distinct messages is interpreted as equality of the UHF outputs. Thus the attacker observes only the zero-output-difference condition

$$F_{\mathbf{k}}(\mathbf{m}) = F_{\mathbf{k}}(\mathbf{m}^*).$$

For a real t-bit PRF-like protector, this idealization can be relaxed quantitatively. If a PH transcript exposes at most q_{PH} distinct protected digest values, accidental collisions among distinct digest values occur with probability at most $\binom{q_{\mathrm{PH}}}{2}2^{-t}$ in the random-function model, and a PRF instantiation adds its distinguishing advantage. PH recovery claims below may therefore be read either in the collision-transparent model or with this additive failure term. The headline NH result uses same-nonce WC(S), where the tested UHF equality is exact and no protector-collision assumption is needed. We refer to the two MAC settings as WC(S) and PH.

Our target is recovery of a reusable KTH hash key through these MAC interfaces. The KTH digest itself is not exposed. To model the post-forgery recovery attacks of [15], we use a persistent interface in which generation and verification remain available after accepted as well as rejected attempts, without rekeying or locking out the attacker. Standard single-forgery games may stop at the first acceptance. They do not model this post-success recovery phase. All query counts below include the accepted verification events used as differential information.

Definition 1 (Controlled forgery). *A controlled differential forgery is a successful MAC forgery for which the attacker knows the two messages and the UHF collision or fixed-difference condition that implies acceptance. An arbitrary forgery with no identified UHF relation is not a controlled differential forgery.*

Lemma 1 (Same-nonce WC(S) forgeries are controlled). *Let* $(\mathbf{m}, N, T)$ *be a valid WC(S) authentication and* $(\mathbf{m}^*, N, T^*)$ *a successful forgery with* $\mathbf{m}^* \neq \mathbf{m}$. *Then*

$$F_{\mathbf{k}}(\mathbf{m}) - F_{\mathbf{k}}(\mathbf{m}^*) = T - T^*.$$

Thus the forgery is controlled, with public difference $\Delta = T - T^*$.

Proof. Subtract the two valid tag equations. Their common mask $g(k', N)$ cancels.

Hence controlledness is automatic for successful same-nonce WC(S) forgeries on distinct messages. The nonce condition is essential: with distinct nonces, $T - T^*$ also contains the unknown mask difference $g(k', N) - g(k', N^*)$. Under the PH assumption above, successful reuse of an authenticated tag on a distinct message similarly exposes the controlled collision $F_{\mathbf{k}}(\mathbf{m}) = F_{\mathbf{k}}(\mathbf{m}^*)$.

2.4 Differential Probability and Image Probability

Definition 2 (Differential probability). *Let* $f : \mathrm{G} \to \mathrm{G}'$ *be a public function. The differential probability of* $(a, b) \in \mathrm{G} \times \mathrm{G}'$ *is*

$$\mathsf{DP}_f(a, b) = \frac{\#\{x \in \mathrm{G} \mid f(x + a) - f(x) = b\}}{\#\mathrm{G}}.$$

We denote $\max_{a\neq 0,b} \mathsf{DP}_f(a,b)$ *by* MDP_f*, and*

$$\max_{a\neq 0} \sum_b \mathsf{DP}_f(a,b)^2$$

by MNDP_f*.*

Definition 3 (Image probability). *Let* $f : \mathrm{G} \to \mathrm{G}'$ *be a public function. The image probability of* $b \in \mathrm{G}'$ *is*

$$\mathsf{IP}(b) = \frac{\#\{x \in \mathrm{G} \mid f(x) = b\}}{\#\mathrm{G}},$$

and $\mathrm{MIP}_f = \max_b \mathsf{IP}(b)$*.*

For bijective f, the Serial and Parallel universality results of [11] are expressed through MDP_f and MNDP_f. The public-function KTH Parallel analyses in [12, 13] include non-bijective fixed-length public functions and image-probability terms. In the setting used below, the variable-length Δ-universality bound is $\max\{\mathsf{MDP}_f, \mathrm{MIP}_f\}$. The Serial-based MAC recovery section below uses the bijective setting. The Parallel section treats both differential tests and the image-probability tests obtained by extending a message by one block. For bijective f, $\mathsf{IP}(b) = 1/\#\mathrm{G}$ for every output value.

2.5 Key-Then-Hash Functions

KTH functions are UHFs of the form

$$F_{\mathbf{k}}(\mathbf{m}) = F(\mathbf{k} + \mathbf{m}),$$

where the function on the right is public and the addition is blockwise over the length of $\mathbf{m}$ [11]. Accordingly, when a vector in G^κ is added to a shorter message, only the prefix matching the message length is used. The message domain is $\mathrm{BS}(\mathrm{G}, \kappa)$, the set of nonempty block strings of lengths $1, \ldots, \kappa$. At each fixed length ℓ, the message space is G^ℓ and is therefore automatically closed under the ℓ-block prefixes of key-space offsets. The offset arguments below are thus local to the message lengths appearing in a test and do not require any additional closure assumption. For two strings $\mathbf{m}, \mathbf{m}^*$ with $|\mathbf{m}| \leq |\mathbf{m}^*|$, their difference is represented by $(\mathbf{a}, \lambda)$, where

$$\mathbf{a} = (m_1 - m_1^*, \ldots, m_{|\mathbf{m}|} - m_{|\mathbf{m}|}^*) \quad \text{and} \quad \lambda = |\mathbf{m}^*| - |\mathbf{m}|.$$

The two generic KTH constructions of [11] are Serial and Parallel. For a message of length ℓ, Serial uses

$$h_0 = 0, \qquad h_i = f(h_{i-1} + k_i + m_i), \qquad \mathrm{Serial}[f]_{\mathbf{k}}(\mathbf{m}) = h_\ell,$$

where $f : \mathrm{G} \to \mathrm{G}$ is a permutation. Parallel uses

$$\mathrm{Parallel}[f]_{\mathbf{k}}(\mathbf{m}) = \sum_{i=1}^{\ell} f(k_i + m_i).$$

For bijective f, Serial is MDP_f-universal and MDP_f-Δ-universal, while Parallel is MNDP_f-universal and MDP_f-Δ-universal [11]. Public-function Parallel bounds are given in [12, 13].

3 No Absolute Weak-Key Classes under Controlled KTH Attacks

Handschuh and Preneel define a weak-key class by unexpectedly favorable attack behavior together with efficient membership detection [15, p. 151]. The formal definition is broad enough that an individual translated solution set may still satisfy it. Our claim is therefore not that such sets cease to exist or cannot be called weak-key classes. The issue is whether the label distinguishes intrinsically exceptional absolute key values. HP08's own design discussion treats that distinction as operational by considering weak-key screening and by proposing regularity as a way to alleviate the concern [15, pp. 145,158]. For a controlled KTH test, conditioning on its solution set makes that test succeed with probability one by definition, so conditional success alone cannot establish exceptional absolute keys. The structural question is whether the same attack behavior is privileged at some absolute key locations. KTH translation preserves class cardinality, uniform-key success probability, message lengths, and oracle-query cost, so every such class has attack-equivalent translates through every absolute key.

3.1 Solution Sets

Definition 4 (Solution set). *For a KTH function F, two messages $\mathbf{m}, \mathbf{m}^*$, and $\Delta \in \mathrm{G}'$, define*

$$\mathcal{S}(\mathbf{m}, \mathbf{m}^*, \Delta) = \{\mathbf{k} \in \mathrm{G}^\kappa : F_{\mathbf{k}}(\mathbf{m}) - F_{\mathbf{k}}(\mathbf{m}^*) = \Delta\}. \tag{1}$$

A successful controlled differential event tells the attacker that the reusable hash key lies in the corresponding solution set. In WC(S), a valid reference tag tests membership with one same-nonce verification query, and Lemma 1 shows that every successful such forgery certifies the tested difference.

3.2 Offset-Invariance

Proposition 1 of [11] states KTH offset-invariance for arbitrary message lengths. The same property gives an exact identity for the keys satisfying a fixed differential equation.

Proposition 1 (Offset-invariance of solution sets). *Let $\mathbf{m}, \mathbf{m}^* \in \mathrm{BS}(\mathrm{G}, \kappa)$ and $\mathbf{o} \in \mathrm{G}^\kappa$. Then*

$$\mathcal{S}(\mathbf{m} + \mathbf{o}, \mathbf{m}^* + \mathbf{o}, \Delta) = \mathcal{S}(\mathbf{m}, \mathbf{m}^*, \Delta) - \mathbf{o}. \tag{2}$$

Here and throughout, adding $\mathbf{o}$ to a message uses the prefix matching that message length. In particular, the two solution sets have the same size.

Proof. A key $\mathbf{k}$ belongs to the left-hand side exactly when

$$F((\mathbf{k}+\mathbf{o})+\mathbf{m}) - F((\mathbf{k}+\mathbf{o})+\mathbf{m}^*) = \Delta,$$

with the relevant key prefix used in each hash input. Set $\mathbf{k}' = \mathbf{k}+\mathbf{o}$. The equation is exactly the condition $\mathbf{k}' \in \mathcal{S}(\mathbf{m}, \mathbf{m}^*, \Delta)$, or equivalently $\mathbf{k} \in \mathcal{S}(\mathbf{m}, \mathbf{m}^*, \Delta) - \mathbf{o}$.

3.3 Solution-Set Size and Universality

For a uniformly sampled key,

$$\Pr_{\mathbf{k}}[\mathbf{k} \in \mathcal{S}(\mathbf{m}, \mathbf{m}^*, \Delta)] = \frac{\#\mathcal{S}(\mathbf{m}, \mathbf{m}^*, \Delta)}{\#\mathrm{G}^\kappa}.$$

Hence ε-Δ-universality gives

$$\#\mathcal{S}(\mathbf{m}, \mathbf{m}^*, \Delta) \le \varepsilon \, \#\mathrm{G}^\kappa \tag{3}$$

for every fixed output difference. For $\Delta = 0$, ordinary ε-universality gives the same bound for collision solution sets.

Theorem 1 (No nontrivial absolute weak-key classes under controlled KTH differential attacks). *Let F be a KTH function on $\mathrm{BS}(\mathrm{G}, \kappa)$ and consider controlled collision or fixed-difference attacks on distinct messages. For every fixed attack $(\mathbf{m}, \mathbf{m}^*, \Delta)$ with a nonempty solution set and for every key $\mathbf{k}_0$, there is a shifted attack with*

1. *the same number and type of oracle queries,*
2. *the same success probability over a uniform key,*
3. *a solution set of the same size, and*
4. *$\mathbf{k}_0$ in that solution set.*

Moreover, ε-universality bounds every collision test by ε. The ε-Δ-universality bound applies to every fixed-difference test.

More strongly, for every offset $\mathbf{o}$, the map

$$(\mathbf{m}, \mathbf{m}^*, \Delta) \longmapsto (\mathbf{m}+\mathbf{o}, \mathbf{m}^*+\mathbf{o}, \Delta)$$

is a bijection from the controlled tests successful under $\mathbf{k}+\mathbf{o}$ to those successful under $\mathbf{k}$. It preserves the message lengths and query type. Thus the entire controlled-test landscape is the same up to translation at every absolute key.

Consequently, these controlled tests give every absolute key an isomorphic attack profile. No key is distinguished by solution-set size, success probability, message lengths, or query type. These attacks therefore do not define a nontrivial absolute weak-key class. The relative size of each solution set in key space is exactly the corresponding collision or fixed-difference probability, and universality bounds that probability.

Proof. Choose any $\mathbf{k}^* \in \mathcal{S}(\mathbf{m}, \mathbf{m}^*, \Delta)$ and set $\mathbf{o} = \mathbf{k}^* - \mathbf{k}_0$. Proposition 1 places $\mathbf{k}_0$ in the shifted solution set and preserves its size. The oracle interface and query count are unchanged. The probability statements follow from the universality definitions and (3). For the final claim, Proposition 1 shows that a test contains $\mathbf{k} + \mathbf{o}$ exactly when its shifted test contains $\mathbf{k}$. Adding $-\mathbf{o}$ gives the inverse map.

Corollary 1 (Absolute-key screening cannot suppress all translates). *Let $\mathcal{K} = \mathrm{G}^\kappa$ and let $S \subseteq \mathcal{K}$ be the solution set of a realizable controlled KTH test. For any fixed probability distribution μ on $\mathcal{K}$,*

$$\mathbb{E}_{\mathbf{o} \leftarrow \mathcal{K}} \left[\Pr_{\mathbf{k} \leftarrow \mu} [\mathbf{k} \in S - \mathbf{o}] \right] = \frac{\#S}{\#\mathcal{K}}.$$

Consequently, some translated realization of the test succeeds under μ with probability at least $\#S/\#\mathcal{K}$. In particular, conditioning uniform key generation on any nonempty screened set cannot make every translate less likely than the original uniform-key density.

Proof. For every fixed $\mathbf{k} \in \mathcal{K}$ and uniform $\mathbf{o}$, the value $\mathbf{k} + \mathbf{o}$ is uniform on $\mathcal{K}$. Hence $\Pr_{\mathbf{o}}[\mathbf{k} \in S - \mathbf{o}] = \#S/\#\mathcal{K}$. Averaging over $\mathbf{k} \leftarrow \mu$ gives the identity. At least one offset attains the average, and Theorem 1 realizes that offset by a translated controlled test with the same message lengths, query type, and oracle-query cost.

Theorem 1 does not invalidate the attacks or their membership tests. The attacks remain valid, but the absolute weak-key diagnosis does not. Every absolute key has an isomorphic Handschuh–Preneel attack profile. For controlled KTH this is a strong form of the regularity question raised in HP08: the attack profile is homogeneous over absolute key space. More precisely, the theorem does not say that all controlled tests have the same probability or that solution sets disappear. It says that any realizable test at one absolute key has, at every other key, a translated counterpart with the same solution-set size, uniform-key probability, message lengths, query type, and oracle-query cost. Hence no absolute location is singled out by the controlled-test family. The invariant is reusable differential structure and the MAC interface, not the absolute key value. Native KTH message spaces are closed under the required offsets at every fixed length. A restricted external message space may block some translations and must be analyzed separately.

Design consequence. Corollary 1 makes the screening consequence distributional: no fixed absolute-key screening rule can suppress every translated realization below the original uniform-key density. Screening can change which translate is favorable, but not remove the translation orbit. The warning against key reuse remains relevant. Before success, universality bounds the relevant collision and fixed-difference probabilities. After an outcome, the useful object is the surviving solution set and its intersections under later tests.

Application to Handschuh and Preneel. The theorem applies to the controlled attacks from [15] realizable on the relevant native KTH domains, including NH, NMH*, WH, and Square Hash. For NH, the equal-subkey class is just $d = 0$ in the partition $k_2 - k_1 = d$. Message changes move the same test through every d. The zero-factor tests similarly move through absolute key guesses. Appendix A gives the concrete derivations.

4 Key Recovery in KTH-Based MACs After a Controlled Differential Forgery

A controlled test still reveals key information: success retains its solution set and failure removes it. We formalize this transcript update before giving generic and affine recovery results.

4.1 Candidate-Set Updates

If the first success has solution set S_0 and preceding failures have solution sets $R_1, \ldots, R_r$, refinement starts from

$$C_0 = S_0 \setminus \bigcup_{i=1}^{r} R_i. \tag{4}$$

For any later test with solution set S', update

$$C \leftarrow C \cap S' \quad \text{on success,} \qquad C \leftarrow C \setminus S' \quad \text{on failure.}$$

Recovery stops when one candidate remains or all available tests agree on the survivors. Computing the solution sets is separate from oracle complexity and may itself require algebraic or differential cryptanalysis.

4.2 Conditional Probability Under Adaptive Testing

Lemma 2 (Conditional success probability). *Suppose previous controlled tests leave a nonempty candidate set C. Conditioned on this transcript, a uniform reusable key is uniform on C. A next test with solution set S' succeeds with probability*

$$\Pr[\text{success} \mid \text{previous outcomes}] = \frac{\#C \cap S'}{\#C}. \tag{5}$$

Proof. Previous answers are exactly the event that the key lies in C; conditioning a uniform key on this event is uniform on C, and the next success is membership in $C \cap S'$.

Thus universality controls an isolated test in the full key space, whereas adaptive recovery is governed by intersections with the surviving set. Since q binary answers yield at most 2^q transcripts, identifying one of n candidates needs at least $\lceil \log_2 n \rceil$ answers in the worst case. Attaining this bound requires balanced tests and does not follow from $\mathsf{MDP}_f < 1$ alone.

4.3 Generic KTH Differential Key Recovery

Fix $\ell \le \kappa$ and let

$$F_\ell : \mathrm{G}^\ell \to \mathrm{G}'$$

denote the public KTH map restricted to ℓ-block inputs. For a nonzero vector difference $\mathbf{a} \in \mathrm{G}^\ell$, define the equal-length derivative

$$D_{\mathbf{a}} F_\ell(\mathbf{x}) = F_\ell(\mathbf{x} + \mathbf{a}) - F_\ell(\mathbf{x}).$$

This is the equal-length generalized KTH differential $(\mathbf{a}, 0, \Delta)$ of [11]. Define

$$\mathsf{DP}_{F_\ell}(\mathbf{a}, \Delta) = \frac{\#\{\mathbf{x} \in \mathrm{G}^\ell : D_{\mathbf{a}} F_\ell(\mathbf{x}) = \Delta\}}{\#\mathrm{G}^\ell}, \qquad \mathsf{MDP}_{F_\ell} = \max_{\mathbf{a} \neq 0,\, \Delta \in \mathrm{G}'} \mathsf{DP}_{F_\ell}(\mathbf{a}, \Delta).$$

Theorem 2 (Generic KTH differential key recovery). *Let $\ell \le \kappa$ and suppose $\mathsf{MDP}_{F_\ell} < 1$. Then every two distinct inputs $\mathbf{x}, \mathbf{y} \in \mathrm{G}^\ell$ can be separated by an equal-length generalized KTH differential: there exist $\mathbf{a} \neq 0$ and $\Delta \in \mathrm{G}'$ such that*

$$D_{\mathbf{a}} F_\ell(\mathbf{x}) = \Delta \qquad \text{and} \qquad D_{\mathbf{a}} F_\ell(\mathbf{y}) \neq \Delta.$$

Consequently, if a controlled forgery involving key prefix $(k_1, \ldots, k_\ell)$ leaves a candidate set $S_0 \subseteq \mathrm{G}^\ell$, an information-theoretically unbounded adversary with access to the corresponding equal-length differential-test oracle can recover that key prefix using at most

$$\#S_0 - 1$$

further oracle answers.

Proof. Let $\mathbf{o} = \mathbf{y} - \mathbf{x} \neq 0$. Since $\mathsf{MDP}_{F_\ell} < 1$, $D_{\mathbf{o}} F_\ell$ is not constant. Hence there exists $\mathbf{z} \in \mathrm{G}^\ell$ such that

$$D_{\mathbf{o}} F_\ell(\mathbf{z}) \neq D_{\mathbf{o}} F_\ell(\mathbf{x}).$$

Put $\mathbf{a} = \mathbf{z} - \mathbf{x} \neq 0$. Commuting finite differences over the abelian group G^ℓ gives

$$D_{\mathbf{a}} F_\ell(\mathbf{y}) - D_{\mathbf{a}} F_\ell(\mathbf{x}) = D_{\mathbf{o}} F_\ell(\mathbf{x} + \mathbf{a}) - D_{\mathbf{o}} F_\ell(\mathbf{x}) \neq 0.$$

Thus $(\mathbf{a}, 0, D_{\mathbf{a}} F_\ell(\mathbf{x}))$ separates the two candidates.

For recovery, while the current candidate set has more than one element, choose any two distinct candidates and an equal-length KTH differential separating them. Whatever the oracle answer, at least one of the pair is removed. Hence at most $\#S_0 - 1$ further answers are required.

Only equal-length differentials are needed for separation, even if an unequal-length forgery produced S_0: refine at the largest key-prefix length on which S_0 depends. The theorem is information-theoretic. Finding separators, representing candidates, and realizing tests through a MAC may be expensive. KTH translation maps the complete transcript for prefix $\mathbf{k}$ to an isomorphic transcript for $\mathbf{k} + \mathbf{o}$, preserving candidate evolution and query count.

Corollary 2 (No post-success absolute weak-key class). *Under the conditions of Theorem 2, every absolute ℓ-block KTH key prefix is differentially recoverable in the information-theoretic setting, and translated prefixes have isomorphic recovery experiments. Differential key recovery therefore does not define a nontrivial absolute weak-key class.*

For permutation-based Serial and Parallel, $\mathsf{MDP}_f < 1$ implies $\mathsf{MDP}_{F_\ell} < 1$ at every fixed length [11]. The public-function Parallel setting has the analogous equal-length bound [12, 13]. We next impose affine structure.

4.4 Affine Recovery from Logarithmically Many Successful Events

Affine derivatives give a structured candidate space and a logarithmic bound on additional *successful* events. Suppose for every $\mathbf{a} \in \mathrm{G}^\ell$,

$$D_{\mathbf{a}} F_\ell(\mathbf{x}) = L_{\mathbf{a}}(\mathbf{x}) + c_{\mathbf{a}}, \tag{6}$$

where $L_{\mathbf{a}} : \mathrm{G}^\ell \to \mathrm{G}'$ is a homomorphism and $c_{\mathbf{a}} = D_{\mathbf{a}} F_\ell(\mathbf{0})$. Constant derivatives and linear structures are classical [18]. A successful test on reference message $\mathbf{m}$, difference $\mathbf{a}$, and output Δ gives

$$L_{\mathbf{a}}(\mathbf{k}_{[1..\ell]}) = \Delta - c_{\mathbf{a}} - L_{\mathbf{a}}(\mathbf{m}), \tag{7}$$

so two candidate prefixes satisfy the same equation iff their difference lies in $\ker L_{\mathbf{a}}$.

Theorem 3 (Logarithmic success count for affine generalized derivatives). *Let $\ell \le \kappa$ and let $F_\ell : \mathrm{G}^\ell \to \mathrm{G}'$ be the public KTH map at length ℓ, with $\#\mathrm{G}^\ell > 1$. Suppose every equal-length derivative of F_ℓ is affine as in* (6) *and $\mathsf{MDP}_{F_\ell} < 1$. Let a retained successful fixed-output generalized differential on reference message $\mathbf{m}_0$, nonzero input difference $\mathbf{a}_0$, and output difference Δ_0 have solution set*

$$S_0 = \{\mathbf{k}' \in \mathrm{G}^\ell : D_{\mathbf{a}_0} F_\ell(\mathbf{k}' + \mathbf{m}_0) = \Delta_0\}.$$

Assume the public description permits efficient computation of the $L_{\mathbf{a}}$, kernel intersections and separating directions below, and efficient solution of the accumulated affine equations. Then S_0 is a coset of $\ker L_{\mathbf{a}_0}$ and

$$\#S_0 = \#\mathrm{G}^\ell \, \mathsf{DP}_{F_\ell}(\mathbf{a}_0, \Delta_0). \tag{8}$$

Moreover, one can efficiently choose at most

$$\lfloor \log_2 \#S_0 \rfloor$$

additional nonzero vector input differences such that one successful fixed-output generalized differential in every chosen direction uniquely determines the unknown ℓ-block KTH key prefix.

Proof. Fix $\mathbf{k}_0 \in S_0$ and put $U_0 = \ker L_{\mathbf{a}_0}$. By the retained affine equation,

$$\mathbf{k}' \in S_0 \iff L_{\mathbf{a}_0}(\mathbf{k}') = L_{\mathbf{a}_0}(\mathbf{k}_0) \iff \mathbf{k}' - \mathbf{k}_0 \in U_0,$$

so

$$S_0 = \mathbf{k}_0 + U_0. \tag{9}$$

Since addition by $\mathbf{m}_0$ is bijective,

$$\#S_0 = \#\{\mathbf{x} \in \mathrm{G}^\ell : D_{\mathbf{a}_0} F_\ell(\mathbf{x}) = \Delta_0\} = \#\mathrm{G}^\ell \, \mathsf{DP}_{F_\ell}(\mathbf{a}_0, \Delta_0),$$

which proves Equation (8).

After additional successful generalized differentials with input differences $\mathbf{a}_1, \ldots, \mathbf{a}_i$, put

$$U_i = U_0 \cap \ker L_{\mathbf{a}_1} \cap \cdots \cap \ker L_{\mathbf{a}_i}. \tag{10}$$

The surviving ambiguity is exactly U_i; recovery is complete when $U_i = \{\mathbf{0}\}$. If not, choose $0 \neq \mathbf{o} \in U_i$. Since $\mathsf{MDP}_{F_\ell} < 1$, $D_{\mathbf{o}} F_\ell$ is not constant, so $L_{\mathbf{o}}$ is nonzero and there exists $\mathbf{a} \in \mathrm{G}^\ell$ with $L_{\mathbf{o}}(\mathbf{a}) \neq 0$. Moreover,

$$\begin{aligned} L_{\mathbf{a}}(\mathbf{o}) &= F_\ell(\mathbf{o} + \mathbf{a}) - F_\ell(\mathbf{o}) - F_\ell(\mathbf{a}) + F_\ell(\mathbf{0}) \\ &= L_{\mathbf{o}}(\mathbf{a}). \end{aligned} \tag{11}$$

Hence $\mathbf{o} \notin \ker L_{\mathbf{a}}$, so $U_i \cap \ker L_{\mathbf{a}}$ is a proper subgroup of U_i and has size at most $\#U_i/2$. Repetition leaves at most $\#S_0/2^s$ candidates after s successful equations; for $s = \lfloor \log_2 \#S_0 \rfloor$ the subgroup is trivial, so the key prefix is unique.

The bound concerns a chosen sequence of successful events, not arbitrary successes. Without the computational assumptions it is only information-theoretic. In the binary settings below, explicit matrices for the $L_{\mathbf{a}}$ reduce kernels, intersections, separating directions, and accumulated equations to Gaussian elimination. Parallel obtains these matrices from its one-block maps, while Xoodoo uses constant-size χ-column systems. The theorem does not claim efficient recovery for an arbitrary KTH map merely because its derivatives are affine.

Corollary 3 (Affine generalized derivatives of Parallel). *Let*

$$\mathrm{Parallel}[f]_\ell(\mathbf{x}) = \sum_{j=1}^{\ell} f(x_j)$$

be fixed-length Parallel. If every derivative of $f : \mathrm{G} \to \mathrm{G}'$ is affine,

$$D_a f(x) = L_a(x) + c_a,$$

then every equal-length generalized derivative of $\mathrm{Parallel}[f]_\ell$ *is affine:*

$$D_{\mathbf{a}}\mathrm{Parallel}[f]_\ell(\mathbf{x}) = \sum_{j=1}^{\ell} L_{a_j}(x_j) + \sum_{j=1}^{\ell} c_{a_j}, \qquad \mathbf{a} = (a_1, \ldots, a_\ell).$$

If additionally $\mathsf{MDP}_f < 1$, then $\mathsf{MDP}_{\mathrm{Parallel}[f]_\ell} < 1$ for every ℓ. Hence Theorem 3 applies directly to any fixed-length Parallel key prefix.

Proof. The displayed derivative follows by componentwise expansion. Its linear part is the homomorphism $(x_1, \ldots, x_\ell) \mapsto \sum_j L_{a_j}(x_j)$. For nonzero $\mathbf{a}$, choose j with $a_j \neq 0$. If the Parallel derivative were constant, fixing all coordinates except x_j would make $D_{a_j} f(x_j)$ constant, contradicting $\mathsf{MDP}_f < 1$.

Corollary 4 (Expected trials). *Let $\mathbf{a}_1, \ldots, \mathbf{a}_s$ be the additional vector input differences obtained from the proof of Theorem 3. For each $\mathbf{a}_i$, choose an output difference Δ_i with positive generalized differential probability. Suppose the authentication interface and KTH construction let the attacker test this fixed output difference. Sample reference messages independently and uniformly from G^ℓ until each differential succeeds. If*

$$p_i = \mathsf{DP}_{F_\ell}(\mathbf{a}_i, \Delta_i),$$

then the expected number of additional differential trials required for all s successes is

$$\sum_{i=1}^{s} \frac{1}{p_i}.$$

In particular, if every selected differential has probability at least p, the expected number of additional trials is at most

$$\frac{\lfloor \log_2 \#S_0 \rfloor}{p}.$$

Proof. For a fixed key prefix, adding an independently uniform reference message makes $\mathbf{k}_{[1..\ell]} + \mathbf{m}$ uniform over G^ℓ. Each trial for $(\mathbf{a}_i, 0, \Delta_i)$ therefore succeeds with probability p_i and needs $1/p_i$ trials in expectation. Linearity of expectation gives the sum. The bound in terms of p follows from $s \leq \lfloor \log_2 \#S_0 \rfloor$ and $p_i \geq p$.

A failed attempt gives an inequality instead of Equation (7) and removes candidates. The theorem and corollary do not use this extra information, so failures may only reduce the stated expected-trial cost. The parameter p is a lower bound on the selected generalized differentials, not the global maximum MDP_f. The latter does not upper-bound the expected trial cost of the proof-selected directions. WC(S) realizes chosen-output equal-length KTH differentials directly, so Corollary 3 can be used on an entire Parallel prefix. Serial under PH instead realizes one-block differentials at non-final blocks through next-block cancellation. Parallel under PH realizes a chosen nonzero branch difference only after one complete branch is known and can supply the compensating difference.

Corollary 5 (First-round equations for iterated permutations). *Consider a KTH function whose public function is an iterated permutation. Suppose the layer before the first non-linear layer is affine and invertible, and suppose the first non-linear layer has affine derivatives. For a fixed external input difference and output difference, every compatible transition at the first non-linear layer gives affine equations on its secret-dependent input state. If these differences determine one first-layer transition, a success reveals those equations directly. If*

several first-layer transitions are compatible, the surviving candidates are contained in the union of their affine solution sets.

This conclusion holds for any number of later rounds. A concrete query complexity additionally requires the probability of the fixed input and output differences and the compatible first-layer transitions.

Proof. The external input difference fixes the difference before the first nonlinear layer through the public affine prefix. For each compatible output difference at that layer, Equation (6) gives affine equations on its input state. Success for the fixed external input and output differences implies that one of the compatible transitions occurred. Later rounds affect which transitions are compatible and how likely this event is, but they do not change the first-layer equations.

5 Single Key-Block Recovery through Serial-Based MACs

Section 4 treated generalized KTH differentials and affine recovery on a whole fixed-length key prefix. Unlike Parallel, however, Serial does not in general inherit affine generalized derivatives from affine derivatives of f because differences propagate through the chaining state. We therefore show how a MAC using Serial with a permutation $f : \mathrm{G} \to \mathrm{G}$, the setting analyzed in [11], localizes a test to one reusable hash-key block. A differential created at one invocation can be canceled before the next invocation, reducing that block to the one-block instance of Theorem 3.

5.1 Localizing One Differential

Consider a reference evaluation and let x_i be the input to the i-th invocation of f. Suppose a modified message changes that input by a. If

$$f(x_i + a) - f(x_i) = b,$$

then the two outputs of the i-th invocation differ by b.

For an interior position $i < \kappa$, put an additional message difference $-b$ in block $i + 1$. This cancels the difference before the next invocation. Because f is bijective, equality from the next invocation onward occurs if and only if the i-th local differential produced exactly b.

For the full-length all-zero reference message, define

$$\mathbf{m} = 0^{\kappa}$$

and, for $i < \kappa$,

$$\mathbf{m}^{(i)}_{a,b} = (0, \ldots, 0, \underbrace{a}_{i}, \underbrace{-b}_{i+1}, 0, \ldots, 0).$$

Then

$$\mathrm{Serial}[f]_{\mathbf{k}}(\mathbf{m}^{(i)}_{a,b}) = \mathrm{Serial}[f]_{\mathbf{k}}(\mathbf{m}) \quad \Longleftrightarrow \quad f(x_i + a) - f(x_i) = b. \tag{12}$$

Thus the full hash comparison is a membership test on the input of one invocation of f.

5.2 WC(S): Reusable-Reference and One-Shot-Nonce Variants

Under the reusable-reference WC(S) variant of Section 2, obtain once

$$T = \mathtt{Sign}(\mathbf{m}, N).$$

The same authenticated reference remains available after every verification answer. For every interior block $i < \kappa$, submitting

$$(\mathbf{m}^{(i)}_{a,b}, N, T)$$

to the verification oracle succeeds exactly when (12) holds. The same signed reference therefore supports all equal-length interior-block tests.

For the final block $i = \kappa$, there is no following block in which to cancel b. Instead, modify only the last block by a. If $f(x_\kappa + a) - f(x_\kappa) = b$, the modified hash is larger than the reference hash by b, so the valid modified tag is

$$T + b.$$

Thus the same reference authentication also supports last-block WC(S) tests. If the verifier instead consumes a nonce after one verification attempt, perform each refinement test with a fresh nonce and a fresh tag on the same reference message. For q refinement tests, the reusable-reference variant costs one generation query and q verification queries, whereas the one-shot-nonce variant costs q generation queries and q verification queries. The tested differential conditions and the number of verification answers needed for recovery are identical.

Fresh random references make the targeted internal input uniform, so finding a fixed differential takes $1/\mathsf{DP}_f(a, b)$ expected trials, each with one generation and one verification query. After a successful reference is retained, later tests use the conditional intersection probabilities of Lemma 2, not fresh full-key-space differential probabilities.

5.3 Recovering the Key Block

For the reference message, let c_{i-1} be the chaining value immediately before block i is added. The input to the i-th public call has the form

$$x_i = c_{i-1} + k_i.$$

Once c_{i-1} is known, the local condition $f(x_i + a) - f(x_i) = b$ is equivalent to

$$c_{i-1} + k_i \in \{x \in \mathrm{G} \mid f(x + a) - f(x) = b\}.$$

Hence the key blocks can be treated sequentially. Recover k_1, compute the next reference chaining value, recover k_2, and continue. The candidate-set updates of Section 4 are applied to a set in G, not to the entire G^κ key space.

5.4 PH and the Last-Block Boundary

In the collision-transparent PH model, the observable event is a full accumulator collision. For $i < \kappa$, the same two-block perturbation $(a, -b)$ makes the full Serial outputs equal exactly when $f(x_i+a)-f(x_i) = b$. Therefore the same localization works for interior key blocks.

For $i = \kappa$, a nonzero input difference at the final invocation of a bijection cannot produce an equal output. The PH localization therefore covers blocks $1, \ldots, \kappa - 1$. Recovering the final block requires an additional mechanism.

6 Key Recovery through Parallel-Based MACs

MACs using Parallel give a second concrete realization of the candidate-set analysis. By Corollary 3, if the public function has affine derivatives then the complete fixed-length Parallel map has affine generalized derivatives as well. Under WC(S), chosen-output equal-length vector differentials are directly observable, so Theorem 3 can recover an entire involved key prefix rather than only one branch. For interfaces that expose less, it is still useful to localize the analysis. Each public-function call is independent before the outputs are combined, and we call each such evaluation a branch. Equal-length messages can give a differential equation on one keyed input, while messages whose lengths differ by one give an image equation on the additional keyed input. The latter mechanism is the key-recovery counterpart of the image-probability term in the variable-length public-function Parallel analyses [12, 13].

6.1 Equal-Length Differential Tests

Use the full-length all-zero reference message. To target block i, change only that block by a:

$$\mathbf{m}_a^{(i)} = (0, \ldots, 0, \underbrace{a}_{i}, 0, \ldots, 0).$$

All other branches cancel in the difference, so

$$\text{Parallel}[f]_{\mathbf{k}}(\mathbf{m}_a^{(i)}) - \text{Parallel}[f]_{\mathbf{k}}(0^\kappa) = f(k_i + a) - f(k_i).$$

Therefore a WC(S) test for output difference b is exactly

$$f(k_i + a) - f(k_i) = b.$$

The reusable-reference and one-shot-nonce query accounting is the same as in Section 5.2. A reusable reference supports all equal-length refinement tests, while the one-shot-nonce variant refreshes the reference for each test.

The same equation can be observed as a PH collision when $b = 0$. For non-bijective f, a one-branch nonzero input difference may therefore give a collision with probability $\mathsf{DP}_f(a, 0)$. For bijective f and $a \neq 0$, this probability is zero,

which is why equal-length PH requires a two-branch starting collision in the bijective case.

Fresh-reference trial accounting is identical to Section 5.2: a fixed local differential needs $1/\mathsf{DP}_f(a, b)$ expected generation/verification pairs, while retained-reference refinement uses conditional candidate-set intersections.

6.2 Unequal-Length Image-Probability Tests

Parallel also permits a direct key-block test that does not use a differential of f. Let $\mathbf{m} = (m_1, \ldots, m_\ell)$ with $\ell < \kappa$, and append one block a:

$$\mathbf{m}^+ = \mathbf{m} \| a.$$

The branches already present in $\mathbf{m}$ are identical in the two evaluations and cancel. Hence

$$\mathrm{Parallel}[f]_{\mathbf{k}}(\mathbf{m}^+) - \mathrm{Parallel}[f]_{\mathbf{k}}(\mathbf{m}) = f(a + k_{\ell+1}). \tag{13}$$

For WC(S), authenticating $\mathbf{m}$ and testing the appropriately shifted tag for $\mathbf{m}^+$ therefore tests the image equation

$$f(a + k_{\ell+1}) = b. \tag{14}$$

For a uniform key block, its success probability is exactly

$$\mathsf{IP}(b) = \frac{\#\{x \in \mathrm{G} : f(x) = b\}}{\#\mathrm{G}}.$$

Choosing an output value attaining MIP_f realizes the largest image-probability test available to WC(S).

A successful test restricts the target key block to the translated preimage

$$k_{\ell+1} \in \{x \in \mathrm{G} : f(x) = b\} - a. \tag{15}$$

Changing the appended block translates the preimage through key space. If f is bijective it is a singleton, so one success determines the target block.

In the collision-transparent PH model, the two messages collide exactly when the additional branch contributes zero:

$$\mathrm{Parallel}[f]_{\mathbf{k}}(\mathbf{m}^+) = \mathrm{Parallel}[f]_{\mathbf{k}}(\mathbf{m}) \quad \Longleftrightarrow \quad f(a + k_{\ell+1}) = 0. \tag{16}$$

Thus PH length extension succeeds with probability $\mathsf{IP}(0)$. For a bijection this is $1/\#\mathrm{G}$ and a success determines the block. The NH result below instead uses the exact WC(S) realization with output difference zero.

Because $\mathrm{BS}(\mathrm{G}, \kappa)$ contains only nonempty strings, the length extension has $1 \le \ell < \kappa$ and directly targets $k_2, \ldots, k_\kappa$. In particular, lengths $\kappa - 1$ and κ target the final block k_κ. This mechanism does not directly target k_1.

With independently uniform appended blocks, each fresh trial has probability $\mathsf{IP}(b)$ under WC(S), or $\mathsf{IP}(0)$ under PH, for every fixed key. After retaining a preimage set, later refinement follows Lemma 2.

Corollary 6 (NH key-pair recovery from WC(S) image tests). *Let one NH multiplication branch have two w-bit keyed inputs and public function*

$$f_{\mathrm{NH}}(x_0, x_1) = x_0 x_1 \bmod 2^{2w}, \qquad x_0, x_1 \in \mathbb{Z}/2^w\mathbb{Z},$$

using their canonical representatives in $\{0, \ldots, 2^w - 1\}$. Then

$$f_{\mathrm{NH}}^{-1}(0) = (\{0\} \times \mathbb{Z}/2^w\mathbb{Z}) \cup (\mathbb{Z}/2^w\mathbb{Z} \times \{0\}),$$

so

$$\mathsf{IP}_{\mathrm{NH}}(0) = \frac{2^{w+1} - 1}{2^{2w}} = 2^{-w+1} - 2^{-2w}.$$

Under reusable-reference WC(S), any NH multiplication pair accessible by length extension can be recovered with one generation query and at most $2^w + 1$ verification queries. Under one-shot-nonce WC(S), the same logical attack uses at most $2^w + 1$ generation queries and $2^w + 1$ verification queries. This is approximately half the verification-query cost per pair in [15, pp. 155–156].

Proof. Let $Q = 2^w$ and write one NH multiplication-pair key as $(k_0, k_1) \in (\mathbb{Z}/Q\mathbb{Z})^2$. For canonical representatives $x_0, x_1 \in \{0, \ldots, Q - 1\}$ we have $0 \le x_0 x_1 < Q^2 = 2^{2w}$. Hence

$$x_0 x_1 \equiv 0 \pmod{2^{2w}} \iff x_0 x_1 = 0 \iff x_0 = 0 \text{ or } x_1 = 0.$$

Thus there are $2Q - 1$ zero preimages, proving $\mathsf{IP}_{\mathrm{NH}}(0) = (2Q - 1)/Q^2$.

Fix enumerations $a_1, \ldots, a_Q$ and $b_1, \ldots, b_Q$ of $\mathbb{Z}/Q\mathbb{Z}$. For reusable-reference WC(S), obtain one authentication $(\mathbf{m}, N, T)$ of a shorter message. For $i = 1, \ldots, Q - 1$, verify the unchanged tag T for $(\mathbf{m} \| (-a_i, -b_i), N)$. Because the same WC(S) nonce mask appears in the reference and verification equations and the tested output difference is zero, acceptance is an exact UHF equality test. No assumption about collisions of a protecting function is needed. By Equation (14) with $b = 0$, verification succeeds exactly when

$$k_0 = a_i \quad \text{or} \quad k_1 = b_i. \tag{17}$$

A failure therefore excludes both a_i and b_i. If all first $Q - 1$ tests fail, the remaining pair is (a_Q, b_Q).

Suppose instead that the first success occurs at index i. Two further tests resolve the disjunction in (17). Choose $b' \neq b_i$ and test $(-a_i, -b')$; conditioned on the original success, this succeeds iff $k_0 = a_i$. Likewise, for any $a' \neq a_i$, testing $(-a', -b_i)$ succeeds iff $k_1 = b_i$. If both succeed the pair is known. If only one succeeds, one word is known and the other has at most $Q - i$ candidates left, because the preceding $i - 1$ failures and the failed disambiguation test exclude the first i candidate values. With one word known, Equation (17) becomes a direct equality test for the other, so at most $Q - i - 1$ more verifications are required. The total is therefore at most

$$i + 2 + (Q - i - 1) = Q + 1 = 2^w + 1.$$

Thus the reusable-reference interface uses one generation query and at most $Q+1$ verification queries. In the one-shot-nonce interface, authenticate the same shorter message under a fresh nonce before each test and use that fresh reference tag once. The attack then uses at most $Q+1$ generation queries and $Q+1$ verification queries, with the same tested equations and verification count.

Handschuh and Preneel first test the Q possible additive relations between the two NH subkeys and then determine one absolute key-word value; their complete-key accounting gives tQ verification queries for t words [15, pp. 155–156], i.e., approximately $2Q$ verifications per multiplication pair. Corollary 6 therefore improves the leading verification-query cost by a factor approaching two.

6.3 Two-Branch Collisions and MNDP

For bijective f, a one-branch nonzero difference between equal-length messages cannot collide. Two unknown branches can instead cancel. Give them nonzero input differences a and a'. If the first produces b, the second must produce $-b$. With independently uniform reference-message blocks, the keyed branch inputs are independent and uniform for every fixed key, so the collision probability is

$$\sum_b \mathsf{DP}_f(a,b)\mathsf{DP}_f(a',-b) \le \mathsf{MNDP}_f.$$

The bound follows from Cauchy–Schwarz and the definition of MNDP_f. Choosing $a' = -a$ gives

$$\mathsf{DP}_f(-a,-b) = \mathsf{DP}_f(a,b),$$

so selecting a direction that maximizes the squared row norm realizes MNDP_f. A success restricts two key blocks to a known joint solution set. It does not reveal b and does not generally determine either block. The unequal-length $\mathsf{IP}(0) = 1/\#\mathrm{G}$ attack instead determines an appended block after one success. The two-branch route supplies complementary starting information, but only the image route generically yields a known branch.

Further tests can refine the two-block set using Section 4, but no generic recovery guarantee follows from one such relation. Once one branch input x is known, it can compensate for a chosen differential on another block. For desired compensating output difference $-b$, bijectivity lets the attacker compute

$$y = f^{-1}(f(x) - b)$$

and use message difference $y - x$ on the known branch. Collision then tests whether the unknown branch produces b. Thus the initial Parallel PH phase is governed by MNDP_f or image probability, while recovery after one known branch returns to ordinary differential probabilities.

WC(S) exposes chosen-output equal-length tests directly. Under PH, Serial uses next-block cancellation, while Parallel starts from zero-output, two-branch, or image events and gains chosen-output tests once one branch is known.

7 Key Recovery through Serial[Xoodoo]-Based MACs

This section specializes Corollary 5 to MACs using a permutation KTH instance. Xoodoo has a 384-bit state arranged as 128 three-bit columns [8]. Its χ layer is quadratic, hence has affine fixed-input-difference derivatives, and is also involutive. The three-bit column structure lets successful first-round transitions yield explicit affine equations on the secret-dependent state.

We then use Xoodoo[3] because published Vortex trails and exact solution counts at this weight connect the external input and output differences to internal χ transitions. The concrete theorem targets the first key block, whose incoming chaining value is $h_0 = 0$. Under PH, the Serial instance must have at least two blocks so that the output difference can be canceled in block two. Later blocks can be attacked sequentially once the preceding key blocks and reference chaining value are known. Xoofff uses six-round Xoodoo, so this is not an attack on Xoofff.

7.1 Differential Propagation Through the Non-Linear Layer χ

A successful differential trial through the MAC interface for fixed input and output differences is a solution-set membership event. Key recovery requires determining what it reveals about the secret-dependent state. Daemen analyzed difference propagation through the shift-invariant mapping χ in his thesis [7, Sect. 6.9.1]. We restate the derivation in our notation and specialize it to Xoodoo's three-bit columns, yielding the linear system used below.

The non-linear layer of Xoodoo is the parallel application of the three-bit mapping

$$\chi(x)_i = x_i + (x_{i+1} + 1)x_{i+2},$$

where $x \in \mathbb{F}_2^3$ and indices are modulo 3. Let $\mathbf{a} = (a_0, a_1, a_2) \neq 0$ be an input difference, and let (b_0, b_1, b_2) be compatible. Since all arithmetic is over $\mathbb{F}_2$, expanding one coordinate gives

$$\begin{aligned}\chi(x + \mathbf{a})_i + \chi(x)_i &= a_i + a_{i+2} + a_{i+1}a_{i+2} + a_{i+2}x_{i+1} + a_{i+1}x_{i+2} \\ &= \chi(\mathbf{a})_i + a_{i+2}x_{i+1} + a_{i+1}x_{i+2}. \end{aligned} \tag{18}$$

Thus, once the input difference is fixed, the derivative of the quadratic map χ is affine in the unknown input column. Writing the three coordinate equations together gives

$$\begin{bmatrix} 0 & a_2 & a_1 \\ a_2 & 0 & a_0 \\ a_1 & a_0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} + \begin{bmatrix} \chi(\mathbf{a})_0 \\ \chi(\mathbf{a})_1 \\ \chi(\mathbf{a})_2 \end{bmatrix}. \tag{19}$$

Write $M(\mathbf{a})$ for the 3×3 matrix on the left-hand side of (19). Its rank counts the independent linear conditions revealed by the transition, while its kernel describes the residual ambiguity between input columns giving the same right-hand side. The next lemma makes both quantities explicit for Xoodoo's three-bit χ.

Lemma 3 (Information from one active column). *For every nonzero* $\mathbf{a} \in \mathbb{F}_2^3$, $M(\mathbf{a})$ *has rank* 2 *and kernel* $\{0, \mathbf{a}\}$. *Hence every compatible active* χ *transition leaves exactly two possible input columns and gives two independent linear equations on* x.

Proof. Multiplying the matrix by $\mathbf{a}$ gives zero, so its rank is at most 2. Choose an index j for which $a_j = 1$. The 2×2 submatrix obtained from the other two rows and columns is $\left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, which has rank 2. The full matrix therefore has rank 2. By rank-nullity, its kernel has dimension one. Since it contains the nonzero vector $\mathbf{a}$, the kernel is $\{0, \mathbf{a}\}$. A compatible system in three variables consequently has $2^{3-2} = 2$ solutions.

In particular, every compatible active column transition has differential probability 2^{-2}.

Lemma 4 (Two different input differences with known affine offsets recover the secret column). *Let* $s \in \mathbb{F}_2^3$ *be a fixed unknown column. For* $j \in \{1, 2\}$, *let* $u_j \in \mathbb{F}_2^3$ *be known and write the actual input column of trial* j *as*

$$x_j = s + u_j.$$

Suppose two successful χ *transitions use distinct nonzero input differences* $\mathbf{a}_1$ *and* $\mathbf{a}_2$. *Let* r_j *denote the known right-hand side of* (19) *for trial* j, *so that*

$$M(\mathbf{a}_j)x_j = r_j.$$

Then the two systems determine s *uniquely, even when* $u_1 \neq u_2$.

Proof. Since u_j is known, each successful transition gives

$$M(\mathbf{a}_j)s = r_j + M(\mathbf{a}_j)u_j.$$

By Lemma 3, the homogeneous ambiguity of the j-th system is $\ker M(\mathbf{a}_j) = \{0, \mathbf{a}_j\}$. The two nonzero input differences are distinct, so

$$\ker M(\mathbf{a}_1) \cap \ker M(\mathbf{a}_2) = \{0\}.$$

The true s satisfies both affine systems, and their combined homogeneous system has trivial kernel. Hence the combined system has the unique solution s.

Lemma 4 lets us solve active first-χ columns independently and invert the public linear prefix afterwards.

7.2 Resolving the Three-Round Differential Trail

The local equations are useful only if the observed input and output differences identify the internal transitions. For every three-round Xoodoo trail core of weight at most 50, Bordes et al. [5, Sects. 6.4 and 7.3] rule out a second trail

with the same input and output differences and prove exact round-differential independence by solution counting. Hence the selected external input and output differences of a weight-36 Vortex trail uniquely determine the first-χ output-difference state and all intermediate differences, and their probability is the product of the three round-differential probabilities.

The public file `DC-Xoodoo-3r.txt` contains four weight-36 records [17]. We use its first such record, uniquely identified by the round-weight profile $12+12+12$. In the file's (x, z) coordinates, the output difference of its first χ layer is

(x,z)	$(0,20)$	$(1,22)$	$(2,7)$	$(2,8)$	$(2,13)$	$(3,14)$
difference	010	001	010	001	100	100.

The immutable source reference in [17] fixes the complete core. A successful trial with the selected external input and output differences therefore gives the input and output differences of every active χ column, so Equation (19) can be applied to those columns. The online query complexity below does not include rediscovering the core.

Selecting two trails from the core. The first χ layer has six active output-difference columns. In the order displayed above, the core requires

$$(010, 001, 010, 001, 100, 100).$$

Writing bit strings as $b_2b_1b_0$, let $\mathcal{A}_v$ be the set of nonzero input differences compatible with output difference v. Proposition 1 of [8] gives

$$\mathcal{A}_{010} = \{010, 011, 110, 111\}, \quad \mathcal{A}_{001} = \{001, 011, 101, 111\},$$
$$\mathcal{A}_{100} = \{100, 101, 110, 111\}.$$

The six input differences at the first χ layer can be chosen independently, so there are $4^6 = 2^{12}$ compatible input-difference patterns. In particular, choose

$$A = (010, 001, 010, 001, 100, 100), \qquad B = (011, 011, 011, 011, 101, 101).$$

Both choices produce the required first-χ output differences, and their input differences are distinct in every active column. Since the public linear prefix before the first χ is invertible, each selected input-difference state at the first χ layer corresponds to a unique external input difference. Corollary 2 of [8] gives weight 2 to every active column, so the first non-linear layer has weight 12.

We also choose the output difference after the final χ layer. In every active column of that layer, let a be the input difference specified by the core and choose the compatible output difference $\chi(a) = D_a\chi(0)$. Applying the final public ρ_{east} layer gives one fixed output difference of Xoodoo[3]. Applying the inverse of the first public linear prefix to A or B gives the corresponding two input differences of Xoodoo[3]. This defines two complete differentials. The selected first transition, the fixed middle transition, and the selected final transition each have weight 12, giving total weight 36. The exact solution count above makes the probability of each differential 2^{-36}.

7.3 Recovering Six Complete Columns

To recover the remaining one-bit ambiguity in each active first-χ column, use the two trails selected above. Write

$$\lambda_{\text{pre}} = \rho_{\text{west}} \circ \theta$$

for the secret-dependent linear part of the first-round prefix before χ. The incoming chaining value, round constant, and chosen message contribute only known affine terms. Thus, for a fixed active column, write the corresponding column of $\lambda_{\text{pre}}(k)$ as s and the known contribution in trial j as u_j. The actual first-χ input is $x_j = s + u_j$.

A successful trial for one of these input and output difference pairs identifies the corresponding first-χ transition and gives the rank-two system $M(\mathbf{a}_j)x_j = r_j$. Removing the known affine contribution gives $M(\mathbf{a}_j)s = r_j + M(\mathbf{a}_j)u_j$. The first successful trail therefore leaves one binary ambiguity. The second trail has a different first-χ input difference in every active column, so Lemma 4 gives the unique s, even if the two successes use different reference messages.

The six active columns are therefore six independent copies of the same three-bit recovery. Applying the local solver to each column recovers

$$6 \cdot 3 = 18$$

bits of the secret-dependent state $\lambda_{\text{pre}}(k)$. No larger linear system is needed at this stage.

7.4 Full-Key-Block Recovery

The six recovered columns cover only part of the state. Xoodoo's cyclic z symmetry supplies the same local problem at translated positions [8, Sect. 5.5]. For $0 \le t < 32$, let τ_t denote cyclic translation by t positions in the Xoodoo z coordinate,

$$\tau_t(x, y, z) = (x, y, (z + t) \bmod 32).$$

Apply τ_t to every state difference of the two selected trails. Differential propagation is invariant under this common z translation, and the round constants cancel in differences. Every translated core still has weight 36, so the exact solution-count result of Bordes et al. applies to it. The fixed constants affect only known affine offsets and do not change consistency or probability. The activity pattern displayed above contains a column at every Xoodoo x coordinate. Applying the same three-bit solver for all 32 translations therefore covers all

$$4 \cdot 32 = 128$$

three-bit columns. These columns assemble the complete secret-dependent pre-χ state

$$Q = \lambda_{\text{pre}}(k).$$

The public map $\lambda_{\mathrm{pre}} = \rho_{\mathrm{west}} \circ \theta$ is invertible, so the original reusable KTH block is obtained directly as

$$k = \lambda_{\mathrm{pre}}^{-1}(Q) = \theta^{-1}\big({\rho_{\mathrm{west}}}^{-1}(Q)\big).$$

The 32 translated pairs form the orbit of the two trails under τ_t. The full recovery is parallel repetition of the three-bit solver followed by one public inverse.

Theorem 4 (Key recovery for one XOODOO[3] **KTH block through MAC oracles).** *Consider the first key block of* Serial[XOODOO[3]]*, whose incoming chaining value is $h_0 = 0$, and the weight-*36 *Vortex core fixed above. Under WC(S), its selected output difference is tested directly when the message ends at this block, or canceled in the following block. Under collision-transparent PH, assume $\kappa \geq 2$ and cancel it in block two as in Section 5. For each of the* 32 *cyclic z translations, use the two complete differentials whose first-χ input differences are different in every active column. Then* 64 *successful differential trials through the MAC interface determine the complete* 384*-bit reusable KTH hash-key block.*

Each targeted differential succeeds with probability 2^{-36} for a uniformly distributed KTH input. An attack using an independently sampled reference message for every attempt therefore requires

$$64 \cdot 2^{36} = 2^{42}$$

expected differential trials. If each trial uses one generation query for the reference message and one verification query for the modified message, this gives 2^{42} queries of each type, or 2^{43} MAC-oracle calls. The count is the same for the reusable-reference and one-shot-nonce WC(S) variants because this implementation authenticates a new reference for every trial. The collision-transparent PH realization has the same generation and verification counts, with the additive protector-collision failure term stated in Section 2. In both settings, the persistent oracle of Section 2 remains available after the 64 *accepted verification events.*

Proof. For every targeted trail, the probability 2^{-36} and the recovery of the internal differences follow from the Vortex weight together with the no-clustering and independence results above. Changing the reference message changes the actual first-χ input only through the known affine contribution u_j. For each active column, the secret-dependent three-bit value s is fixed across trials, and the two successful trails recover it by the line-intersection argument of Lemma 4. The same local computation is applied independently to the six active columns and to all 32 cyclic translations. These translations cover all 128 columns and therefore determine $Q = \lambda_{\mathrm{pre}}(k)$. Applying $\lambda_{\mathrm{pre}}^{-1}$ gives the KTH hash-key block used by the MAC. Finally, every trial with an independently sampled reference message has success probability 2^{-36} for every fixed key, so the expected number of attempts for each required success is 2^{36}, and the attack needs 64 such successes.

The count uses independently sampled references. Reusable-reference dependence requires separate analysis. After one block is recovered, its chaining value is known, so the attack proceeds sequentially under WC(S), and through block $\kappa - 1$ under PH. The final PH block remains outside the cancellation interface.

8 Conclusion

Handschuh and Preneel were right that key reuse can turn apparently limited UHF forgeries into devastating key-recovery attacks. For the KTH families and controlled tests studied here, however, the attacks do not expose exceptional absolute keys. They expose translated differential solution sets. KTH offset invariance makes the complete controlled-test landscape isomorphic at every absolute key. In this controlled KTH setting, that is a strong form of the regularity sought by Handschuh and Preneel. Here regularity means homogeneity of the controlled attack profile over absolute key space, not the absence of differential solution sets. The attacks survive, while the absolute weak-key diagnosis does not.

This distinction gives a positive design rule. For this controlled attack family, choose the public function and set the hash-key reuse policy according to the differential and image probabilities underlying the applicable universality bounds, rather than according to a screened subset of "good" absolute keys. Screening particular values cannot remove the phenomenon because the same test shape translates through every key. After success, recovery is governed instead by the surviving solution set and its intersections. Handschuh and Preneel's warning about key reuse therefore remains fully relevant.

The corrected viewpoint is also productive. The NH image attack under reusable-reference WC(S) recovers an addressable multiplication pair with one generation query plus at most $2^w + 1$ verification queries, about half the verification queries of Handschuh–Preneel. In general, no probability-one nonzero differential implies pairwise separation of key-prefix candidates. Affine generalized derivatives turn suitably selected successful events into enough affine equations to determine the prefix. For Serial[XOODOO[3]], published trails instantiate this mechanism and recover a reusable 384-bit KTH block in 2^{42} expected differential trials, or 2^{43} MAC-oracle calls.

For KTH-based MAC design, forgery analysis and post-forgery key-recovery analysis should therefore be treated as two views of the same controlled differential structure.

Acknowledgements

Generative AI use. ChatGPT (GPT-5.6 Sol, OpenAI) was used interactively to revise author-developed drafts and ideas, improve exposition, organization, and presentation, and discuss the clarity and consistency of technical arguments. AI-suggested material was reviewed and edited by the author, who takes full responsibility for the final manuscript.

References

1. Armour, M., Cid, C.: Partition oracles from weak key forgeries. In: CANS 2021. LNCS, vol. 13099, pp. 42–62. Springer (2021). https://doi.org/10.1007/978-3-030-92548-2_3, also Cryptology ePrint Archive, Report 2021/1296
2. Bellare, M., Canetti, R., Krawczyk, H.: Keying Hash Functions for Message Authentication. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1109, pp. 1–15. Springer (1996). https://doi.org/10.1007/3-540-68697-5_1
3. Bernstein, D.J.: Stronger Security Bounds for Wegman-Carter-Shoup Authenticators. In: Cramer, R. (ed.) Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3494, pp. 164–180. Springer (2005). https://doi.org/10.1007/11426639_10
4. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and Secure Message Authentication. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 216–233. Springer (1999). https://doi.org/10.1007/3-540-48405-1_14
5. Bordes, N., Daemen, J., Kuijsters, D., Van Assche, G.: Thinking Outside the Superbox. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12827, pp. 337–367. Springer (2021). https://doi.org/10.1007/978-3-030-84252-9_12
6. Cui, T., Grassi, L.: Algebraic key-recovery attacks on reduced-round xoofff. In: Selected Areas in Cryptography - SAC 2020. Lecture Notes in Computer Science, vol. 12804, pp. 171–197. Springer (2021). https://doi.org/10.1007/978-3-030-81652-0_7
7. Daemen, J.: Cipher and hash function design, strategies based on linear and differential cryptanalysis, PhD Thesis. K.U.Leuven (1995), available from https://cs.ru.nl/~joan/JoanDaemenPapers.html
8. Daemen, J., Hoffert, S., Van Assche, G., Van Keer, R.: The design of Xoodoo and Xoofff. IACR Trans. Symmetric Cryptol. **2018**(4), 1–38 (2018). https://doi.org/10.13154/tosc.v2018.i4.1-38
9. Dunkelman, O., Weizman, A.: Another look at differential-linear attacks. In: Selected Areas in Cryptography, SAC 2022. Lecture Notes in Computer Science, vol. 13742, pp. 116–136. Springer (2024). https://doi.org/10.1007/978-3-031-58411-4_6
10. Etzel, M., Patel, S., Ramzan, Z.: SQUARE HASH: fast message authentication via optimized universal hash functions. In: Wiener, M.J. (ed.) Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Lecture Notes in Computer Science, vol. 1666, pp. 234–251. Springer (1999). https://doi.org/10.1007/3-540-48405-1_15
11. Fuchs, J., Rotella, Y., Daemen, J.: On the security of keyed hashing based on public permutations. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology

- CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part III. Lecture Notes in Computer Science, vol. 14083, pp. 607–627. Springer (2023). https://doi.org/10.1007/978-3-031-38548-3_20
12. Ghosh, K., Amiri-Eliasi, P., Daemen, J.: Multimixer-128: Universal keyed hashing based on integer multiplication. IACR Trans. Symmetric Cryptol. **2023**(3), 1–24 (2023). https://doi.org/10.46586/TOSC.V2023.I3.1-24
13. Ghosh, K., Fuchs, J., Amiri-Eliasi, P., Daemen, J.: Universal hashing based on field multiplication and (near-)mds matrices. In: Mrabet, N.E., Feo, L.D., Duquesne, S. (eds.) Progress in Cryptology - AFRICACRYPT 2023 - 14th International Conference on Cryptology in Africa, Sousse, Tunisia, July 19-21, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14064, pp. 129–150. Springer (2023). https://doi.org/10.1007/978-3-031-37679-5_6
14. Halevi, S., Krawczyk, H.: MMH: software message authentication in the gbit/second rates. In: Biham, E. (ed.) Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20–22, 1997, Proceedings. Lecture Notes in Computer Science, vol. 1267, pp. 172–189. Springer (1997). https://doi.org/10.1007/BFb0052345
15. Handschuh, H., Preneel, B.: Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In: Wagner, D.A. (ed.) Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5157, pp. 144–161. Springer (2008). https://doi.org/10.1007/978-3-540-85174-5_9
16. Kaps, J.P., Yüksel, K., Sunar, B.: Energy scalable universal hashing. IEEE Transactions on Computers **54**(12), 1484–1495 (2005). https://doi.org/10.1109/TC.2005.195
17. Keccak Team: DC-Xoodoo-3r.txt. KeccakTeam/Xoodoo, commit 37de9b9 (2021)
18. Lai, X.: Additive and linear structures of cryptographic functions. In: Fast Software Encryption, FSE 1994. Lecture Notes in Computer Science, vol. 1008, pp. 75–85. Springer (1995). https://doi.org/10.1007/3-540-60590-8_6
19. Luykx, A., Preneel, B.: Optimal forgeries against polynomial-based MACs and GCM. In: EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 445–467. Springer (2018). https://doi.org/10.1007/978-3-319-78381-9_17
20. Mella, S., Daemen, J., Van Assche, G.: Tighter Trail Bounds for Xoodoo. IACR Trans. Symmetric Cryptol. **2023**(4), 187–214 (2023). https://doi.org/10.46586/tosc.v2023.i4.187-214, https://eprint.iacr.org/2022/1088
21. Minematsu, K., Tsunoo, Y.: Provably secure MACs from differentially-uniform permutations and AES-based implementations. In: Fast Software Encryption, FSE 2006. Lecture Notes in Computer Science, vol. 4047, pp. 226–241. Springer (2006). https://doi.org/10.1007/11799313_15
22. Nandi, M.: Bernstein bound on WCS is tight: Repairing luykx–preneel optimal forgeries. In: CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 213–238. Springer (2018). https://doi.org/10.1007/978-3-319-96881-0_8
23. Procter, G., Cid, C.: On weak keys and forgery attacks against polynomial-based MAC schemes. In: Moriai, S. (ed.) Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8424, pp. 287–304. Springer (2013). https://doi.org/10.1007/978-3-662-43933-3_15

24. Shoup, V.: On Fast and Provably Secure Message Authentication Based on Universal Hashing. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1109, pp. 313–328. Springer (1996). https://doi.org/10.1007/3-540-68697-5_24
25. Stinson, D.R.: Universal Hashing and Authentication Codes. In: Feigenbaum, J. (ed.) Advances in Cryptology - CRYPTO '91. Lecture Notes in Computer Science, vol. 576, pp. 74–85. Springer (1991). https://doi.org/10.1007/3-540-46766-1_5
26. Stinson, D.R.: On the Connections Between Universal Hashing, Combinatorial Designs and Error-Correcting Codes. Electron. Colloquium Comput. Complex. **2**(52) (1995), https://eccc.weizmann.ac.il/eccc-reports/1995/TR95-052/index.html
27. Wang, P., Li, Y., Zhang, L., Zheng, K.: Related-key almost universal hash functions: Definitions, constructions and applications. In: Fast Software Encryption, FSE 2016. Lecture Notes in Computer Science, vol. 9783, pp. 514–532. Springer (2016). https://doi.org/10.1007/978-3-662-52993-5_26
28. Wegman, M.N., Carter, L.: New Hash Functions and Their Use in Authentication and Set Equality. J. Comput. Syst. Sci. **22**(3), 265–279 (1981). https://doi.org/10.1016/0022-0000(81)90033-7
29. Zheng, K., Wang, P.: A uniform class of weak keys for universal hash functions. Cryptology ePrint Archive, Report 2017/436 (2017), https://eprint.iacr.org/2017/436
30. Zhou, H., Li, Z., Dong, X., Jia, K., Meier, W.: Practical key-recovery attacks on round-reduced ketje jr, xoodoo-AE and xoodyak. The Computer Journal **63**(8), 1231–1246 (2020). https://doi.org/10.1093/comjnl/bxz152
31. Zhou, H., Zong, R., Dong, X., Jia, K., Meier, W.: Interpolation attacks on round-reduced elephant, kravatte and xoofff. Cryptology ePrint Archive, Paper 2020/781 (2020), https://eprint.iacr.org/2020/781

A The Handschuh–Preneel KTH Families

We spell out NH and Square Hash, while NMH* and WH follow the same local mechanism. The underlying UHF sources are [4, 10, 14, 16].

NH. For one NH pair [4], $f(x_1, x_2) = x_1 x_2 \pmod{2^{2w}}$, with each keyed input word formed modulo 2^w. Writing $X = m_1 + k_1$ and $Y = m_2 + k_2$, changing the message pair by $(a, -a)$ swaps the two keyed inputs exactly when $k_2 - k_1 = a + m_1 - m_2$. The product is then unchanged. The condition $k_1 = k_2$ is only one choice of the right-hand side. The divide-and-conquer attack of [15, pp. 155–156] tests arbitrary additive relations between the two key words. As the relation value ranges over $\mathbb{Z}/2^w\mathbb{Z}$, the sets

$$\{(k_1, k_2) : k_2 - k_1 = d\}$$

partition the two-word key space into 2^w equal parts. The class $d = 0$ is not weaker than the others. A purported absolute weak-key classification must either privilege one value of d without cryptographic justification or include every value

of d and hence every key. Every part has relative size 2^{-w}, exactly the scale given by NH's universality bound [4].

Handschuh and Preneel also give the zero-factor attack: when one keyed factor is zero, changing the other factor leaves the pair product unchanged. This directly tests a key-word value. KTH message addition moves the zero input to any key guess, so the behavior again belongs to the public multiplication rather than to a special absolute key value.

NMH and WH.* NMH* uses inner additions modulo 2^w [15, p. 152] and is KTH over $G = \mathbb{Z}/2^w\mathbb{Z}$, with later operations absorbed into the public function. Handschuh and Preneel group NMH, NH, and WH under the same zero-factor and additive-relation attacks [15, pp. 155–156]. WH uses the additive group of $\mathrm{GF}(2^w)$ [16]. In both cases message offsets translate the tested relation, so equality of two subkeys is not privileged.

Square Hash. Let $q = 2^w$. In its native word-oriented form, Square Hash [10] is $\mathrm{SQH}_{\mathbf{k}}(\mathbf{m}) = \left[\sum_i ((m_i + k_i) \bmod q)^2\right] \bmod p$, for $m_i, k_i \in \mathbb{Z}_q$. Its KTH input group is $G = \mathbb{Z}/q\mathbb{Z}$. Handschuh and Preneel apply their equal-subkey Proposition 2 to Square Hash [15, pp. 152,155]. More generally, if $k_j - k_i = d \pmod q$, replacing (m_i, m_j) by $(m_j + d, m_i - d) \pmod q$ swaps the two keyed inputs and leaves the hash unchanged. As d ranges over $\mathbb{Z}_q$, the classes $\{(k_i, k_j) : k_j - k_i = d\}$ partition the two-word key space into q equal parts. Their equal-subkey case is $d = 0$, so no additive relation is privileged.