On hashing with tweakable ciphers

This item was submitted to Loughborough University's Institutional Repository by the author.


Additional Information:

- This is a conference paper [© IEEE]. It is also available at: http://ieeexplore.ieee.org/ Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Metadata Record: https://dspace.lboro.ac.uk/2134/5684

Version: Published

Publisher: © IEEE

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
On Hashing With Tweakable Ciphers

Raphael C.-W. Phan
Loughborough University, UK
Email: R.Phan@lboro.ac.uk

Jean-Philippe Aumasson
FHNW, Windisch, Switzerland
Email: JeanPhilippe.Aumasson@gmail.com

Abstract—Cryptographic hash functions are often built on block ciphers in order to reduce the security analysis of the hash to that of the cipher, and to minimize the hardware size. Well known hash constructs are used in international standards like MD5 and SHA-1. Recently, researchers proposed new modes of operations for hash functions to protect against generic attacks, and it remains open how to base such functions on block ciphers. An attracting and intuitive choice is to combine previous constructions with tweakable block ciphers. We investigate such constructions, and show the surprising result that combining a provably secure mode of operation with a provably secure tweakable cipher does not guarantee the security of the constructed hash function. In fact, simple attacks can be possible when the interaction between secure components leaves some additional “freedom” to an adversary. Our techniques are derived from the principle of slide attacks, which were introduced for attacking block ciphers.

I. INTRODUCTION

Cryptographic hash functions are key ingredients in numerous schemes like public-key encryption, digital signatures, message-authentication codes, or multiparty functionalities. During the last few years, the focus on hash functions has dramatically increased, because of new dedicated attacks on e.g. MD5 and SHA-1, and new generic attacks—that is, which apply to broad classes of functions. A hash function $h$ should satisfy (at least)

- **collision resistance**: it should be hard to find distinct inputs $x$ and $x'$ such that $h(x) = h(x')$
- **second-preimage resistance**: given a random input $x$, it should be hard to find a distinct $x'$ such that $h(x) = h(x')$
- **preimage resistance**: given $h(x)$ for a random unknown $x$, it should be hard to find a distinct $x'$ such that $h(x) = h(x')$

Critical generic attacks [1]–[3] were presented against the classical Merkle-Damgård (MD) iterative mode of operation, thus threatening all the hash functions using the MD operation mode (for example, MD5 and SHA-1). An MD hash function $H$ hashes a message $M = M_1 M_2 \ldots M_t$ as follows: for $1 \leq i \leq t$, compute

$$h_i = f(h_{i-1}, M_i),$$

where $f$ is called the **compression function**, and $h_0$ is a pre-defined initialization vector (IV). Finally, the function returns the hash value $H(M) = h_t$.

To prevent from attacks on the MD mode, extended operation modes were proposed (e.g. HAIFA [4], [5]); in this work we focus on Rivest’s **dithered MD** (DMD) mode [6], [7], for its simplicity and better efficiency. DMD was proposed as a general framework for hash functions, and it remains to be seen how to concretely instantiate the underlying compression function. Furthermore, we know of no concrete hash function construction that employs DMD. The question has been discussed among the community as to whether the upcoming NIST hash function competition [8] should concentrate on only concrete hash function proposals, or split between proposals for operating modes and for compression functions (cf. [9]).

Block cipher-based constructions for hash functions used to build on the MD mode [10], [11] (the so-called PGV schemes) and there is no direct way to extend them to DMD, because of an additional input to the compression function. Recently, ad-hoc efficient constructions were analyzed [12], but it is unclear whether this approach is optimal. A natural approach suggested by Rivest [6], [7] is to use **tweakable block ciphers** [13] to instantiate DMD hash functions. The model of tweakable block cipher was originally proposed by Liskov, Rivest, and Wagner to define families of permutations for a fixed secret key, thus avoiding the slowdown caused by the key schedule operation.

A. Contribution

We consider two classes of constructions for DMD hash functions based on tweakable block ciphers, which combine

1) a secure hash mode of operation
2) a secure tweakable block cipher
3) a secure block cipher-based construction

Then, we show that such constructions do not necessarily lead to a secure hash function. More precisely, we apply the idea of slid pairs to find collisions for one of the function classes. Our attacks apply to broad classes of constructions, and are independent of the strength of the underlying block cipher used.

B. Related Work

Dithering of hash functions appeared with the work of Kelsey and Schneier [1], with generalizations in [4]–[7]. An analysis of dithered hash functions appears in [14], and constructive results were proposed in [12].

Hash functions based on block ciphers recently attracted considerable attention, with several results proving security...
bounds for constructions with one or more block ciphers [15]–[18]. Concrete block cipher-based designs include Maelstrom [19] and Grindahl [20], and implicitly the de facto standards MD5, SHA-1, and SHA-2.

The first known application of the sliding techniques [21]–[24] to the context of hash functions was by [25]; namely to slide the compression function of SHA-1. This result was extended in [26], and later applied to the SHA based block cipher SHACAL-1. In [27], a slide-style attack was presented on incremental hash functions based on pair block chaining. Recently, Gorski et al. applied sliding to mount distinguishing attacks on Sage type hash functions [28]; as well as key recovery attacks on Sage hash function based MACs. The generic hash function attacks by Dean [29], and Kelsey and Schneier [1] exploit a fixed-point of the compression function, which is similar in spirit to slide attacks [21], [22] and to our attacks. Schneier [1] exploit a fixed-point of the compression function, which is similar in spirit to slide attacks [21], [22] and to our attacks. Our results are the first known work that slide dithered hash functions.

In [30], attacks were mounted on a hash function mode based on tweakable block ciphers, so called Tweak Chain Hash (TCH). This construction, however, is a conventional hash function (not dithered). The attack on TCH does not apply to the constructions we consider in this paper.

II. DEFINITIONS

A block cipher is a map $E : \{0,1\}^k \times \{0,1\}^m \mapsto \{0,1\}^m$, such that $E_K(\cdot) = E(K, \cdot)$ is a permutation of $\{0,1\}^m$ for all $K \in \{0,1\}^k$, and its inverse permutation is written $E^{-1}$. The set of all block ciphers with $k$-bit key and $m$-bit messages is denoted $\text{Bloc}(k, m)$. A block cipher-based hash function is a map $H : \text{Bloc}(k ,m) \times D \mapsto R$, where $D \subseteq \{0,1\}^r$ and $R = \{0,1\}^n$, defined iteratively by a compression function $f : \text{Bloc}(k ,m) \times \{0,1\}^{n_1} \times \{0,1\}^{n_2} \mapsto \{0,1\}^{n_2}$, where $n_1$ is the size of a message block, and $n_2$ the size of chaining values. In the remainder of the paper, we assume $m = n_1 = n_2 = n$.

A. Hash Compression Function Modes based on Block Ciphers

Among the 12 provably secure hash compression function modes presented in [10], we will focus on the most popular ones (which are used in all concrete hash designs):

- the Matyas-Meyer-Oseas [31] (MMO) mode, which constructs the compression function by setting $h_i = E_{h_{i-1}}(M_i) \oplus M_i$.
- the Davies-Meyer mode, somehow the dual of MMO, is used in the MD5 and SHA functions:
  $$h_i = E_{M_i}(h_{i-1}) \oplus h_{i-1}.$$ 
- the Miyaguchi-Preneel mode [32], [33], notably employed in Whirlpool [34], where the underlying block cipher is a variant of Rijndael:
  $$h_i = E_{h_{i-1}}(M_i) \oplus h_{i-1} \oplus M_i.$$ 

B. Tweakable Block Ciphers

Tweakable block ciphers [13] aim to achieve the “best of both worlds” (security and efficiency) for block cipher-based hashing, since they allow to use an input-dependent permutation, thus avoiding the time-consuming key schedule—the additional input (tweak) being injected in a simplistic fashion.

Formally, a tweakable block cipher has three inputs: a key $K \in \{0,1\}^k$, a tweak $T \in \{0,1\}^t$ and a message $M \in \{0,1\}^n$; it produces a ciphertext $C \in \{0,1\}^n$:

$$E : \{0,1\}^k \times \{0,1\}^t \times \{0,1\}^n \mapsto \{0,1\}^n.$$ 

We write $E_K(T, M)$ as shorthand for $E(K, T, M)$. In [13], two tweakable block ciphers are constructed from conventional block ciphers:

- **TEXE** (tweakable cipher formed by two $E$ boxes sandwiching an XOR), which is inspired from CBC-MAC and is defined as
  $$E_K(T, M) = E_K(T \oplus E_K(M)).$$ 

It was proven [13] that TEXE is a secure tweakable cipher in the sense of indistinguishability from a family of random permutations parametrized by the tweak.

- **TFX** (inspired from the generalized DESX construction FX of Kilian and Rogaway [35], [36]) defines the scheme
  $$E_K(T, M) = E_K(M \oplus U(T)) \oplus U(T),$$

where $U$ is a universal hash function TFX is strongly (chosen-ciphertext) secure in the sense of indistinguishability from a family of random permutations parametrized by the tweak (see [13] for details).

III. DITHERED HASH FUNCTIONS (DMG)

Dithering is a generalization of the countermeasure proposed by Kelsey and Schneier [1] to prevent attacks [1], [29] based on message block repetition and fixed-points. This type of iterated hash uses a sequence of dither values $D = d_1 \ldots d_\ell$, which is public and static.

A dithered Merkle-Damgård (DMG) hash function, as defined in [6], [7], takes as input an IV, a message $M = M_1M_2 \ldots M_\ell$, a dither sequence $D = d_1d_2 \ldots d_\ell$, and produces an output $H_D(M)$ as follows: for $1 \leq i \leq \ell$, compute

$$h_i = f(h_{i-1}, M_i, d_i)$$

where $f$ is the compression function, the $h_i$’s are chaining variables; $h_0$ is the IV and the dither values $d_0, \ldots, d_{\ell-1}$ have a zero most significant bit (MSB), and $d_\ell$, the last dither value, has nonzero MSB.

In the above definition, a special MSB encoding of $d_i$ differentiates the last block from other blocks. This feature was proposed to avoid a complex message padding rule (unlike classical MD functions, which append to a message the encoding of its bit length).
IV. DMD CONSTRUCTIONS

We consider constructions of DMD hash functions where the compression function is instantiated with a tweakable block cipher, in one of the 12 provably secure PGV modes [11]. The dither input $d_i$ of the compression function is directed to the tweak input $T$ of the tweakable block cipher as suggested in [6], [7]. We focus on the MMO mode to simplify the description, though our results equally apply to other modes as detailed in Section V-A.

A. DMD-TEXE in MMO Mode

This construction of a DMD function combines the TEXE and MMO schemes, which respectively add a new input slot and construct a secure compression function. Following our above definitions, DMD-TEXE with the MMO scheme defines

$$h_i = f(h_{i-1}, M_i, d_i)$$

where $f$ is an arbitrary function, not necessarily invertible.

$$h_{i+1} = E_{h_{i-1}}(d_i, M_i) + M_i$$

$$h_{i+1} = E_{h_{i-1}}(d_i + E_{h_{i-1}}(M_i)) + M_i.$$

This construction, however, is inefficient, since each call to the compression function requires two encryptions with the block cipher $E$, plus one key schedule (both encryptions use the same key).

B. DMD-TFX in MMO Mode

This construction is more efficient than DMD-TEXE, since it requires only one encryption (and the key schedule), plus a call to a universal hash function, which is generally faster than the block cipher in practice. This construction defines:

$$h_i = f(h_{i-1}, M_i, d_i)$$

$$h_{i+1} = E_{h_{i-1}}(d_i, M_i) + M_i$$

$$h_{i+1} = E_{h_{i-1}}(d_i + M_i) + M_i.$$

Eventually we have $h = h_{i+1}$ and $h' = h'_i$, and thus $H_D(M) = H_D(M')$, i.e. a collision.

A. Generalization

This attack generalizes to other DMD-TFX in PGV modes, for example when the underlying block cipher-based compression function sets

$$E_{h_{i-1}}(M_i + h_i - 1) + M_i + h_i - 1,$$

or

$$E_{h_{i-1}}(M_i + h_i - 1) + M_i.$$

These are the PGV mode constructions called $f_2$ and $f_4$ in [11]. Our attack also applies to the PGV mode popularly known as Miyaguchi-Preneel.

More generally, our attack can in general be applied to DMD hash functions with symmetric mixing of the message block and the dither input. Examples of this kind would be inspired from perceptual hash functions in image authentication applications [37], [38] of the field of image processing, from which the notion of “dithering” is inspired. In such hash functions, the dither input is mixed within the compression function in the same way as how a message input is mixed, hence the mixing is symmetric.

We can describe a further generalization, when the compression function $f$ can be expressed as

$$h_i = f(h_{i-1}, M_i, d_i)$$

$$h'_i = f'(h_{i-1}, f_1(M_i) \otimes f_2(d_i))$$

where $\otimes$ and $f_1$ and $f_2$ are arbitrary invertible functions, and $f_2$ is an arbitrary function, not necessarily invertible.

Our attack can be applied to this case too, by choosing $M'_i$ such that

$$f_1(M'_i) \otimes f_2(d_i) = f_1(M_{i+1}) \otimes f_2(d_{i+1}),$$

i.e., we choose:

$$M'_i = f_1^{-1} \{ f_1(M_{i+1}) \otimes f_2(d_{i+1}) \}.$$

Note that $f_1$ and $f_2$ can be defined to also take the chaining variable $h_{i-1}$ as input, and our attack equally applies.

B. Applicability to DMD-TEXE Functions

Functions based on the TEXE construction resist our attacks since the mixing between $M_i$ and $d_i$ is not invertible, with respect to expressing $M_i$ from the above equation in terms of the other variables $h_i, h_{i-1}$ and $d_i$.

It might seem counter-intuitive that DMD is insecure against our attack when instantiated with TFX yet is resistant when
instantiated with the essentially weaker [13] TEXE. This might
be because the difference between the two security notions
achieved by TFX and TEXE is in terms of the access to the
decryption oracle, which does not appear to be useful since
PGV modes only make use of the underlying block cipher in
the encryption direction.

VI. CONCLUSIONS

 Attacks such as [1], [29] seem to provide support for the
hypothesis [6], [7], [39] that the adversary attacking an MD
hash function has too much control over the message block
input to the compression function, and so this control should
be restricted [1], [6], [7], e.g. with dithering which furthermore
makes the process of hashing a message block dependent on
its position within the entire message [39], or alternatively
makes the compression function round-dependent. DMD hash
functions therefore increase the security guarantees, by using
different compressions of the message at each iteration.

 However, care should be taken when using dithers for this
restriction, since a dither input, although is predefined, is
another input channel and another degree of freedom that an
adversary could potentially exploit; as our attacks showed.

 Viewed from another perspective, even if two copies of the
hash iteration are out of phase thus leading to differences
between the copies due to differing round positions and hence
different dither inputs, an adversary can still use message block
differences to shift the position dependence, and subsequently
offset them.

 A countermeasure for our attack is to append to the message
its bit length; this makes the function less efficient and in
fact clashes with the DMD design objective of achieving
efficiency and of eliminating the need for conventional mes-
 sage padding. It remains open, though, whether concrete hash
functions should be based on block ciphers, or be dedicated
designs in order to satisfy the very particular security re-
quirements of a hash function, not necessarily captured by
those of block ciphers (cf. notions like indifferentiability, seed-
 incompressibility, secure MAC, etc.).

Biham [4] highlights an important point in the design and
analysis of hash functions: that the same technology, principles
and design criteria as block ciphers should be used for hash
functions. He further explains that this is supported by the
fact that the block cipher design criteria of strong avalanche
criterion (SAC) and strong diffusion, ensures that neutral bits
 (used to attack hash functions) cannot exist; and a good
example is the case of MDx and SHAx hash functions that
have slow diffusion which probably explains why it is easier
to control differences through their rounds than it is to do so
for block ciphers. Furthermore, it is well known that block
 ciphers and hash functions can be converted to each other
through some suitable mode of operation. To this purpose, we
have:

 1) considered the security of the dithered MD hash function
constructions and how its security is affected by its inter-
actions with the underlying tweakable block cipher and
corresponding block cipher-based compression function mode.

 2) shown how to apply the block cipher cryptanalytic
 technique of slide attacks to mount collision attacks on
hash functions.

For the first point, our results show that caution needs to be
exercised when interacting primitives even if they are provably
secure: in this case, the secure tweakable cipher TFX and the
secure MMO (resp. Miyaguchi-Preneel, f_2, f_3) modes.

For the second point, we contrast between a block cipher
and a hash function in terms of the difficulty of applying the
kind of attacks treated in this paper. In some PGV modes of
operation, the round key input K of the block cipher is the
message block input M_i of the hash function, and while there
exists a key schedule in block ciphers, hash functions either
have no schedule to process message blocks or the schedule is
linear (e.g. SHA). Therefore, round keys going into each round
of the block cipher depend on a master key via a complex
key schedule, and thus the round keys are difficult to control,
whereas the message blocks to the hash function can be easily
controlled by qan attacker.

Furthermore, an adversary attacking a block cipher has no
control over the intermediate states of the cipher, unless one
considers the context of related-key attacks, which even in that
case only gives restricted control in the sense that an adversary
may be able to differentially affect the round keys of two or
more copies of the encryption iteration but yet not know let
alone be able to control what the round key values are. In
contrast for the case of hash functions, the message blocks
that are in some PGV modes the analog of round keys in
block ciphers, are not only known but can be chosen by the
adversary. All these points indicate that it should be easier
to mount sliding techniques to hash functions.

Slide attacks on block ciphers require some form of period-
icty for the encryption or decryption iteration, while for hash
functions we demonstrated that this is no longer necessary
because the difference due to sliding different rounds can
be offset by the mixing of message block differences and
dither inputs. And while the dithering mechanisms proposed
in [6], [7] prevent the attacks in [1], [29] by eliminating
message block repetitions, our attacks apply even if there are
no repetitions in the sequence.

Our work further supports the view [39] that the role that
sequences play in iterated cryptographic constructions needs
to be further studied. In this particular case, nonrepetitive
sequences would complicate slide attacks for block ciphers
but apparently not necessarily for hash functions.

REFERENCES

for much less than 2^n work.” in EUROCRYPT, ser. LNCS, R. Cramer,


[3] ——, “Herding hash functions and the Nostradamus attack.” in EURO-
183–200.