Collisions are not Incidental: A Compression Function Exploiting Discrete Geometry

Dimitar Jetchev[1], Onur Özen[1], and Martijn Stam[2]

[1] Laboratory for Cryptologic Algorithms, EPFL, Switzerland
[2] Department of Computer Science, University of Bristol, UK

Abstract. We present a new construction of a compression function $H\colon \{0,1\}^{3n} \to \{0,1\}^{2n}$ that uses two parallel calls to an ideal primitive (an ideal blockcipher or a public random function) from $2n$ to n bits. This is similar to the well-known MDC-2 or the recently proposed MJH by Lee and Stam (CT-RSA'11). However, unlike these constructions, we show already in the compression function that an adversary limited (asymptotically in n) to $\mathcal{O}(2^{2n(1-\delta)/3})$ queries (for any $\delta > 0$) has disappearing advantage to find collisions. A key component of our construction is the use of the Szemerédi–Trotter theorem over finite fields to bound the number of full compression function evaluations an adversary can make, in terms of the number of queries to the underlying primitives. Moveover, for the security proof we rely on a new abstraction that refines and strenghtens existing techniques. We believe that this framework elucidates existing proofs and we consider it of independent interest.

1 Introduction

Ever since the initial efforts to turn a blockcipher into a hash function, a major drawback of using blockcipher-based compression functions producing a digest size equal to the block-length is that the digest size is too small to produce a hash function meeting today's security requirements. For example, AES, operating on 128 bits, limits collision resistance to at most 2^{64} operations/queries. As a remedy, *double-length* compression functions and corresponding double-length hash functions have been introduced (e.g. [3–5]): A design that outputs $2n$ bits (while making several calls to a blockcipher with n-bit blocks) could potentially provide collision resistance up to roughly 2^n blockcipher evaluations.

In this work, we are interested in the construction of a provably collision-resistant (beyond $2^{n/2}$ queries) compression function from $3n$ to $2n$ bits making two parallel calls to an ideal primitive from $2n$ to n bits (either a public random function– PuRF–or an ideal blockcipher with n-bit blocks and n-bit keys). Our motivation is a natural one: All existing designs in this class fall short. There is no proof, or the proof is not known to extend to the blockcipher case; (non-trivial) collision resistance is only provided in the iteration; the primitive calls need to

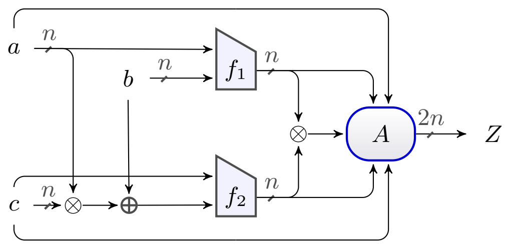

Fig. 1. Our compression function $H^{f_1,f_2}\colon \{0,1\}^{3n} \to \{0,1\}^{2n}$ illustrated (see Construction 1 for the details).

be made in sequence; or the number of calls is higher. Yet known impossibility bounds [11,12,15] give no reason why such a construction should not be possible.

Our Contribution. We provide a construction (see Fig. 1), for which we prove that any adversary limited (asymptotically in n) to $\mathcal{O}(2^{2n(1-\delta)/3})$ queries (for any $\delta > 0$) has disappearing advantage to find collisions. To the best of our knowledge, this is the first design of its kind offering collision resistance beyond $2^{n/2}$ queries. Our construction has two key innovative components (see Fig. 1): a preprocessing function C^{PRE} that transforms the $3n$-bit input into a pair of $2n$-bit strings that are passed as inputs to the two ideal primitive calls; and a postprocessing function C^{POST} that combines the two outputs of the ideal primitives and the $3n$-bit input into the $2n$-bit output of the compression function. Initially, we will concentrate on the PuRF scenario; details for the more complicated ideal-cipher model follow later (Section 6). In either case, we work in the ideal-primitive model (giving separate proofs for each scenario).

A major technical hurdle in the proof of collision resistance is that the standard proof techniques turn out to be insufficient. For concreteness, consider an adversary that adaptively makes three queries trying to create a collision. Customarily, one would upper bound the probability p_i (for $i = 1, 2, 3$) that an adversary causes a collision on the ith query, say with $B_i = 1/4$ each; taking a union bound leads to an overall bound $3/4$. Our first abstraction is a game hop where we allow an adversary to choose its success probability p_i directly, rather than computing it based on which query to some primitive is being made. By requiring $p_i \leq B_i$ this leads to the same overall winning bound $3/4$, achieved by a greedy adversary. However, this abstraction allows us to phrase and study different scenarios as well (relevant for our collision resistance proof), for instance one where we only set a global requirement $\sum_i p_i \leq 1/2$. Now potentially each of the p_i values could be $1/2$ itself, so using $\sum_i B_i$ would lead to an overall bound of $3/2$ (which is vacuous for a probability), yet intuitively no adversary should be able to do better than $1/2$. A further complication arises when we require the adversary to obtain a success at least twice. While it is easy to deal with non-adaptive adversaries, properly taking care of adaptive adversaries is non-trivial. We provide the abstraction and solutions to the problems just described in Section 3. We believe this framework to be of independent theoretical interest.

The main innovation of our design is the choice for C^{PRE}: the $3n$-bit input is transformed into a pair of an affine line on $\mathbb{F}_{2^n}^2$ and a point on that line. Hence, any given valid input pair to the underlying ideal primitives corresponds to an incidence between a point and a line in the affine plane $\mathbb{F}_{2^n}^2$ over the finite field $\mathbb{F}_{2^n}$. We then use a classical result of discrete geometry, the Szemerédi–Trotter theorem over finite fields, to bound the number of incidences between a set of q lines and a set of q points on $\mathbb{F}_{2^n}^2$, namely by roughly $q^{3/2}$.

The postprocessing is inspired by the Rogaway–Steinberger construction [10], where a special type of $\mathbb{F}_{2^n}$ linear map is used. However, we add the product of the two primitive-outputs to the inputs to this linear map. This turns out to be crucial for our collision resistance proof. In Section 5 we prove that the best strategy for any collision-finding adversary is (close to) maximizing the number of the aforementioned point-line incidences (in C^{PRE}). Our proof uses the newly developed techniques given in Section 3 to deal with adaptive adversaries.

Putting the pieces together, we achieve the claimed collision resistance of already at the compression function level. We also prove (everywhere) preimage resistance up to $\mathcal{O}(2^{(1-\delta)n})$ queries (for arbitrary $\delta > 0$). From an efficiency perspective, our construction makes two parallel calls to distinct primitives, each with $2n$-bit inputs. The overhead consists of a number of xors (to implement the matrix-multiplication) plus, more significantly, two full ($\mathbb{F}_{2^n}$) finite field multiplications: one during the preprocessing and one during the postprocessing.

2 Preliminaries

Primitive-Based Compression Functions. A *compression function* is a map $H\colon \{0,1\}^{tn} \to \{0,1\}^{sn}$, where n is an integer (the block-length, which in an asymptotic setting typically takes the role of the security parameter) and $t > s > 0$ are integer parameters. A compression function is *primitive-based* if it is computed by a program making calls to a finite number of specified oracles (primitives). We use superscripts to denote oracle access. For integers c and n, let $\text{Func}(cn, n)$ denote the set of all maps $\{0,1\}^{cn} \to \{0,1\}^n$ and let $f \overset{\$}{\leftarrow} \text{Func}(cn, n)$ denote that f is sampled uniformly at random from all elements in $\text{Func}(cn, n)$. Then we call f a *public random function* (PuRF) and we refer to a compression function making oracle calls to f as *PuRF-based*. For given input W we denote the resulting digest as $H^f(W)$. More generally, when there are r independently sampled primitives $f_1, \ldots, f_r \overset{\$}{\leftarrow} \text{Func}(cn, n)$ we write $H^{f_1,\ldots,f_r}(W)$.

Similarly, let $\text{Block}((c-1)n, n)$ denote the set of all blockciphers having $(c-1)n$-bit key and operating on n-bit blocks. In other words, $\text{Block}((c-1)n, n)$ is the set of all maps $E\colon \{0,1\}^{(c-1)n} \times \{0,1\}^n \to \{0,1\}^n$, such that for any key $K \in \{0,1\}^{(c-1)n}$, $E(K, \cdot)$ is a permutation on the set $\{0,1\}^n$. (Note that $(c-1)n + n = cn$, so that one can interpret $E \in \text{Func}(cn, n)$ as well.) For a blockcipher E, we denote its inverse by D, so for all $K \in \{0,1\}^{(c-1)n}$ and

$X \in \{0,1\}^n$ we have that $D(K, E(K, X)) = X$. When $E \xleftarrow{\$} \text{Block}((c-1)n, n)$ is chosen uniformly at random we call it an *ideal cipher* and refer to a compression function H^E (or more generally, $H^{E_1,\ldots,E_r}$ when there are r independently sampled blockciphers $E_1, \ldots, E_r \xleftarrow{\$} \text{Block}((c-1)n, n)$) as *blockcipher-based.* The definitions and the illustrations below are provided in the PuRF-based setting; the blockcipher-based case is analogous, where we assume that oracle access to E implicitly implies access to its inverse D as well.

We study *single-layer* compression functions. This means that the oracle calls can be made in parallel and the output of the compression function is computed based on the results of these calls, as well as on the input itself. Formally, let $C_i^{\text{PRE}}\colon \{0,1\}^{tn} \to \{0,1\}^{cn}$ for $i = 1, \ldots, r$, and $C^{\text{POST}}\colon \{0,1\}^{tn} \times (\{0,1\}^n)^r \to \{0,1\}^{sn}$, be pre and postprocessing functions, respectively. Given a tn-bit input W, compute output $Z = H^{f_1,\ldots,f_r}(W)$ as follows: for $i = 1, \ldots, r$, let $x_i \leftarrow C_i^{\text{PRE}}(W)$ and $y_i \leftarrow f_i(x_i)$; the output is then $Z \leftarrow C^{\text{POST}}(W, y_1, \ldots, y_r)$.

Security Notions. An adversary is an algorithm (typically modelled as an interactive Turing machine) that uses its oracle access to the underlying primitives of the compression function in order to 'break' some well-defined property. We will limit ourselves to (everywhere) preimage resistance and collision resistance, and consider information-theoretic adversaries only; our sole resource of interest is the number of queries made to their oracles (adversaries are considered computationally unbounded). Without loss of generality, adversaries are assumed not to repeat queries nor to query an oracle outside of its specified domain.

When, for some $l \in \{1, \ldots, r\}$, an adversary makes an f_l-query obtaining $Y = f_l(X)$, we will append (l, X, Y) to the *query history* $\mathcal{Q}$ (which is initialized empty). For preimage and collision resistance, adversarial success can be determined based on the query history $\mathcal{Q}$ only, which we formalize using the yield set (Definition 1) and which we exploit by dropping the explicit sampling of the primitives f_i and the queries $\mathcal{Q}$ for experiments. We partition $\mathcal{Q}$ in $\mathcal{Q}[1] \ldots \mathcal{Q}[r]$ depending on which of the primitives was called and, although technically elements of $\mathcal{Q}$ are triples, we assume that the context suffices to determine which of the r primitives was used. For $i \leq |\mathcal{Q}|$, we let $\mathcal{Q}_i$ denote the first i elements of $\mathcal{Q}$. Occasionally, we abuse notation by writing $X \in \mathcal{Q}$ or $Y \in \mathcal{Q}$.

Definition 1. *Let $H^{f_1,\ldots,f_r}$ be a primitive-based compression function and let $\mathcal{Q}$ be a set of queries (with answers) to the underlying primitives, then the yield set* $\text{yieldset}(\mathcal{Q})$ *is the set of all pairs (W, Z) such that $Z = H^{f_1,\ldots,f_r}(W)$ and all queries necessary for the evaluation of the compression function at W are in $\mathcal{Q}$. We refer to the cardinality of* $\text{yieldset}(\mathcal{Q})$ *as the* yield *and denote it by* $\text{yield}(\mathcal{Q})$. *Additionally, we define* $\text{yield}(q) = \max_{\mathcal{Q}} \text{yield}(\mathcal{Q})$ *where $|\mathcal{Q}[i]| \leq q$. (Note that since $\mathcal{Q}$ incorporates the primitives' answers, the maximum implicitly includes a maximization over the choice of the underlying primitives.)*

Definition 2 (Collision resistance). *Let $H^{f_1,\ldots,f_r}$ be a primitive-based compression function. For a given $\mathcal{Q}$ and $Z \in \{0,1\}^{sn}$, define*

$$\mathsf{coll}(\mathcal{Q}) \equiv \exists_{Z, W \neq W'} (W, Z), (W', Z) \in \text{yieldset}(\mathcal{Q}) \ .$$

The collision-finding advantage of an adversary $\mathcal{A}$ is defined as

$$\mathsf{Adv}_H^{\mathsf{coll}}(\mathcal{A}) = \Pr\left[f_1...f_r \overset{\$}{\leftarrow} \mathrm{Func}(cn,n), \mathcal{Q} \leftarrow \mathcal{A}^{f_1\cdots f_r} \,:\, \mathsf{coll}(\mathcal{Q})\right] \ .$$

Similarly, define $\mathsf{Adv}_H^{\mathsf{coll}}(q) = \max_{\mathcal{A}} \mathsf{Adv}_H^{\mathsf{coll}}(\mathcal{A})$, *where the maximum is taken over all adversaries $\mathcal{A}$ making at most q queries to each of the underlying primitives.*

Definition 3 (Everywhere preimage resistance). *Let $H^{f_1,\ldots,f_r}$ be a primitive-based compression function. For a given $\mathcal{Q}$ and $Z \in \{0,1\}^{sn}$, define*

$$\mathsf{epre}_Z(\mathcal{Q}) \equiv \exists_{W'}(W', Z) \in \mathrm{yieldset}(\mathcal{Q}) \ .$$

The everywhere preimage-finding advantage of an adversary $\mathcal{A}$ is defined as

$$\mathsf{Adv}_H^{\mathsf{epre}}(\mathcal{A}) = \max_{Z\in\{0,1\}^{sn}} \left\{\Pr\left[f_1...f_r \overset{\$}{\leftarrow} \mathrm{Func}(cn,n), \mathcal{Q} \leftarrow \mathcal{A}^{f_1\cdots f_r} \,:\, \mathsf{epre}_Z(\mathcal{Q})\right]\right\} \ .$$

We also define $\mathsf{Adv}_H^{\mathsf{epre}}(q) = \max_{\mathcal{A}} \mathsf{Adv}_H^{\mathsf{epre}}(\mathcal{A})$, *where the maximum is taken over all adversaries $\mathcal{A}$ making at most q queries to each of the r primitives.*

3 Probabilistic Analysis of Adaptive Adversaries

Most of the security proofs in the literature for compression and hash functions rely on the same principle. Consider the game depicted in Fig. 2, where the adversary has access to some underlying primitive $f()$ and tries to set a predicate $\mathbf{E}$ that is defined for all collections of query-response pairs. We are primarily interested in monotone predicates $\mathbf{E}$, that once set cannot be 'unset' by additional queries. A predicate $\mathbf{E}$ is monotone if and only if for all $\mathcal{Q} \subseteq \mathcal{Q}'$ it holds that $\mathbf{E}(\mathcal{Q}) \Rightarrow \mathbf{E}(\mathcal{Q}')$. Additionally, we impose non-triviality of the predicate meaning that the predicate is not set from the outset (i.e. $\mathbf{E}(\emptyset) = \texttt{false}$). For collision resistance, one should read coll (Definition 2) for $\mathbf{E}$ and for preimage resistance epre_Z (Definition 3). Note that coll and epre_Z are always monotone and that, for our construction, both coll and epre_Z are non-trivial.

Bounding an advantage is then tantamount to bounding $\Pr[\mathbf{E}(\mathcal{Q})]$, where the probabilities are taken over the randomness of f and the coins of $\mathcal{A}$, if any. In the following, we show how we can analyse such events in a stepwise approach to determine useful upper bounds.

There is a distinction between adaptive and non-adaptive adversaries. The latter are required to commit to a fixed set of queries at the very beginning of the game. In the information-theoretic setting, it is customary (and WLOG) to consider deterministic adversaries only. Consequently, maximizing over all q-query (non-adaptive) adversaries becomes equivalent to maximizing over all possible query sets of cardinality q. This considerably simplifies proofs. For instance, when providing a proof in the ideal-cipher model (using a union bound), for a non-adaptive adversary every response can be considered *fully* random, whereas for an adaptive adversary previous queries to the cipher might influence the outcome slightly.

```
$\mathsf{Exp}^{\mathbf{E}\text{-ad}}(\mathcal{A})$:
    Let $i \leftarrow 0, \mathcal{Q}_0 \leftarrow \emptyset$
    While $i < q$ do
        $i \leftarrow i+1$
        $x_i \leftarrow \mathcal{A}(\mathcal{Q}_{i-1})$
        $y_i \leftarrow f(x_i)$
        $\mathcal{Q}_i \leftarrow \mathcal{Q}_{i-1} \cup \{(x_i, y_i)\}$
    Return $\mathbf{E}(\mathcal{Q}_q)$.

$\mathsf{Exp}^{\mathbf{E}\text{-na}}(\mathcal{A})$:
    $(x_1, \ldots, x_q) \leftarrow \mathcal{A}()$
    Let $i \leftarrow 0, \mathcal{Q}_0 \leftarrow \emptyset$
    While $i < q$ do
        $i \leftarrow i+1$
        $y_i \leftarrow f(x_i)$
        $\mathcal{Q}_i \leftarrow \mathcal{Q}_{i-1} \cup \{(x_i, y_i)\}$
    Return $\mathbf{E}(\mathcal{Q}_q)$.

$\mathsf{Exp}^{\mathbf{E},\mathbf{F}}(\mathcal{A})$:
    Let $i \leftarrow 0, \mathcal{Q}_0 \leftarrow \emptyset$
    While $i < q$ do
        $i \leftarrow i+1$
        $x_i \leftarrow \mathcal{A}(\mathcal{Q}_{i-1})$
        $y_i \leftarrow f(x_i)$
        $\mathcal{Q}_i \leftarrow \mathcal{Q}_{i-1} \cup \{(x_i, y_i)\}$
    Return $\mathbf{E}(\mathcal{Q}_q) \wedge \neg\mathbf{F}(\mathcal{Q}_q)$.
```

Fig. 2. Standard adaptive ($\mathsf{Exp}^{\mathbf{E}\text{-ad}}(\mathcal{A})$) and non-adaptive ($\mathsf{Exp}^{\mathbf{E}\text{-na}}(\mathcal{A})$) security games for predicate $\mathbf{E}$, as well as the flagged experiment $\mathsf{Exp}^{\mathbf{E},\mathbf{F}}(\mathcal{A})$.

Related work. Maurer [7] (see also Pietrzak [8]) developed a methodology to equate adaptive and non-adaptive adversaries in certain cases. While it is possible to phrase our game from Fig. 2 in their framework, for many of our winning predicates adaptive adversaries *do* have an advantage over non-adaptive adversaries. Instead we opt for a more direct approach, where we primarily take our inspiration from existing hash-function security proofs. Henceforth, unless stated otherwise, we will consider adaptive adversaries only (and consequently drop the "ad" suffix in naming experiments and advantages).

The straightforward approach. The standard way of dealing with adaptive adversaries, as exemplified for instance by the security proofs [1, 2, 13] for the PGV compression functions [9], is the following. Suppose an adversary makes q queries. These are necessarily made in sequence, so denote with $\mathcal{Q}_i$ the set of query-responses after i queries have been made (where $i \in \{0, \ldots, q\}$). The overall winning probability can then be stated as a sum of the probability of winning on the ith step, where these 'stepwise' probabilities are only taken over the choice of y_i. This makes derivation of the overall bound relatively easy (even when taking into account the accompanying maximization).

Proposition 1. *Let* $\mathbf{E}$ *be a monotone non-trivial predicate. Then the advantage of any (adaptive) adversary* $\mathcal{A}$ *playing* $\mathsf{Exp}^{\mathbf{E}}(\mathcal{A})$ *(see Fig. 2) is bounded by*

$$\mathsf{Adv}^{\mathbf{E}}(\mathcal{A}) \leq \sum_{i=1}^{q} \max_{\mathcal{Q}_{i-1} s.t. \neg\mathbf{E}(\mathcal{Q}_{i-1})} \max_{x_i} \Pr\left[\mathbf{E}(\mathcal{Q}_i) \mid \mathcal{Q}_{i-1} \wedge x_i\right] \ .$$

Using an auxiliary flag. Although easy, the standard approach has the disadvantage that for many more involved constructions, the maximum probalities can get too large. This is typically due to the maximum being attained only for relatively obscure values for $\mathcal{Q}_i$, values that themselves are extremely unlikely to occur. To weed out these unwanted cases, the analysis is often enhanced by splitting the monotone predicate into a set of auxiliary events. For some positive integer k, let $\mathbf{E}_1, \ldots, \mathbf{E}_k$ be predicates such that (for all $\mathcal{Q}$) $\mathbf{E}(\mathcal{Q}) \Rightarrow \bigvee_i^k \mathbf{E}_i(\mathcal{Q})$, then a union bound implies $\Pr[\mathbf{E}] \leq \sum_i^k \Pr[\mathbf{E}_i]$. Several examples of proofs using auxiliary events can be found in the realm of double-length hash functions [6, 14].

```
$\mathsf{Exp}^{\mathbf{B}}(\mathcal{A})$:
    Let $i \leftarrow 0$
        While $i < q$ do
        $i \leftarrow i + 1$
        $p_i \leftarrow \mathcal{A}()$
        if $0 \leq p_i \leq B_i$ then
            with probability $p_i$ return true
    Return false .
```

```
$\mathsf{Exp}^{\mathsf{B}_\Sigma}(\mathcal{A})$:
    Let $i \leftarrow 0$
    While $i < q$ do
        $i \leftarrow i + 1$
        $p_i \leftarrow \mathcal{A}()$
        if $0 \leq \sum_{j=1}^{i} p_j \leq B_\Sigma$ then
            with probability $p_i$ return true
    Return false .
```

Fig. 3. Game-playing interpretation of the adaptive security game (where $\mathbf{B} = (B_1, \ldots B_q)$) and our refined abstract flagging game.

The events $\mathbf{E}_i(\mathcal{Q})$ themselves are usually composed as the conjunction of a monotone event and a *negated* monotone event. In the simplest scenario, consider a second (non-trivial) monotone predicate $\mathbf{F}$. If we define $\mathbf{E}_1 = \mathbf{E} \wedge \neg\mathbf{F}$ and $\mathbf{E}_2 = \mathbf{F}$ then $\mathbf{E} \Rightarrow \mathbf{E}_1 \vee \mathbf{E}_2$ is satisfied. To bound $\Pr[\mathbf{E}_2] = \Pr[\mathbf{F}]$ we can use Proposition 1; for $\Pr[\mathbf{E}_1] = \Pr[\mathbf{E} \wedge \neg\mathbf{F}]$ Proposition 2 shows how the use of the predicate $\mathbf{F}$ effectively allows us to consider a more restricted class of $\mathcal{Q}_i$.

Proposition 2. *Let $\mathbf{E}$ be a non-trivial monotone predicate and let $\mathbf{F}$ be an arbitrary auxiliary non-trivial monotone predicate. Then the advantage of (adaptive) adversary $\mathcal{A}$ setting $\mathbf{E} \wedge \neg\mathbf{F}$ is bounded by*

$$\Pr[\mathbf{E} \wedge \neg\mathbf{F}] \leq \sum_{i=1}^{q} \Pr[\mathbf{E}(\mathcal{Q}_i) \mid \neg\mathbf{E}(\mathcal{Q}_{i-1}) \wedge \neg\mathbf{F}(\mathcal{Q}_{i-1})] \leq \sum_{i=1}^{q} B_i \ ,$$

where $B_i = \max_{\mathcal{Q}_{i-1} s.t. \neg\mathbf{E}(\mathcal{Q}_{i-1}) \wedge \neg\mathbf{F}(\mathcal{Q}_{i-1})} \max_{x_i} \Pr[\mathbf{E}(\mathcal{Q}_i) \mid \mathcal{Q}_{i-1} \wedge x_i]$.

An alternative interpretation. We now make a far bigger step, removing most of the underlying mechanics of the original game. Instead of letting the adversary output elements x_i and then determining by virtue of y_i whether the adversary wins this round, we directly bound the latter probability. That is, in experiment $\mathsf{Exp}^{\mathbf{B}}(\mathcal{A})$ we let the adversary output a probabilities p_i and imagine that $\mathbf{E}$ is set with probability p_i. To avoid this game becoming vacuous (namely if the adversary would output some $p_i = 1$) we put bounds B_i and B'_i on the adversary's success probability. These bounds correspond to the actual game: they are the highest possible success probabilities any adversary in any run can achieve in round i. These probabilities are reminiscent of the conditional probabilities used in the derivation from the standard approach and indeed we can formalize this relationship. Since in $\mathsf{Exp}^{\mathbf{B}}(\mathcal{A})$ a straightforward application of the union bound leads to an overall upper bound of the winning probability of $\sum_{i=1}^{q} B_i$, we can recover Proposition 2.

Lemma 1. *Consider games $\mathsf{Exp}^{\mathbf{E},\mathbf{F}}$ and $\mathsf{Exp}^{\mathbf{B}}$ and subject to*

$$B_i = \max_{\mathcal{Q}_{i-1}\ s.t.\ \neg\mathbf{E}(\mathcal{Q}_{i-1}) \wedge \neg\mathbf{F}(\mathcal{Q}_{i-1})} \max_{x_i} \Pr[\mathbf{E}(\mathcal{Q}_i)] \ \text{. Then,}$$

for all adversaries $\mathcal{A}$, there exists an adversary $\mathcal{A}'$ s.t. $\mathsf{Adv}^{\mathbf{E},\mathbf{F}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathbf{B}}(\mathcal{A}')$.

3.1 A More Refined Approach

In $\mathsf{Exp}^{\mathbf{B}}(\mathcal{A})$ instead of the guards $0 \leq p_i \leq B_i$ for all i, we could have used $0 \leq \sum_{j=1}^{i} p_j \leq \sum_{j=1}^{i} B_j$ as well. With a seemingly minor modification, this leads to another, different game where instead of bounding step-specific by $\sum_{j=1}^{i} B_j$, we always use the same bound B_Σ, as in the game $\mathsf{Exp}^{\mathsf{B}_\Sigma}(\mathcal{A})$ (Fig. 3).

Proposition 3. *For any adversary $\mathcal{A}$, it holds that* $\mathsf{Adv}^{\mathsf{B}_\Sigma}(\mathcal{A}) \leq B_\Sigma$.

Usage. The $\mathsf{Exp}^{\mathsf{B}_\Sigma}$ game captures a special kind of condition that one can encounter in the $\mathsf{Exp}^{\mathbf{E},\mathbf{F}}$ game. For any given $\mathcal{Q}$, we can *a posteriori* determine the probabilities of success by taking $\mathcal{Q}_{i-1}$ and x_i from $\mathcal{Q}$ and then looking at the probability that a *freshly drawn* y_i causes a $\mathbf{E}$. The overall *a posteriori* probability of some $\mathcal{Q}$ is the sum (over i) of these probabilities. The maximum attainable probability this way determines B_Σ, as formalized in Lemma 2. Of note here is the observation that for certain games the B_Σ obtained here is much smaller than the $\sum_{i=1}^{q} B_i$ one would obtain from application of Lemma 1. Very broadly speaking (and with some abuse of notation), it is the difference between $\max_{\mathcal{Q}} \{\sum_i p_i(\mathcal{Q})\}$ and $\sum_i \max_{\mathcal{Q}} \{p_i(\mathcal{Q})\}$.

Lemma 2. *Consider the game* $\mathsf{Exp}^{\mathbf{E},\mathbf{F}}$. *For any given $\mathcal{Q}$, define*

$$p_i(\mathcal{Q}) = \begin{cases} 0, & \text{if } \mathbf{E}(\mathcal{Q}_{i-1}) \vee \mathbf{F}(\mathcal{Q}_{i-1}) \text{ or } |\mathcal{Q}| < i \\ \Pr\left[y_i \leftarrow f(x_i) \,:\, \mathbf{E}(\mathcal{Q}_{i-1} \cup \{(x_i, y_i)\}) \mid \mathcal{Q}_{i-1}\right] & \text{otherwise} \end{cases}$$

and let $B_\Sigma = \max_{\mathcal{Q}} \sum_{i=1}^{q} p_i(\mathcal{Q})$. Then for all adversaries $\mathcal{A}$ there exists an adversary $\mathcal{A}'$ such that $\mathsf{Adv}^{\mathbf{E},\mathbf{F}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{B}_\Sigma}(\mathcal{A}')$.

3.2 Counting Successes

In the previous games we considered a predicate $\mathbf{E}(\mathcal{Q})$ that could either be `true` or `false` . In other words, we were interested in at least one success occurring. In some scenarios, counting the number of succcesses is more appropriate. To this end, let ctr be a function such that $\mathsf{ctr}(\mathcal{Q}) \in \mathbb{N}$ and $\mathsf{ctr}(\mathcal{Q}_i) - \mathsf{ctr}(\mathcal{Q}_{i-1}) \in \{0, 1\}$ for all possible $\mathcal{Q}_i$. For future reference, define the event $\mathsf{hit}(\mathcal{Q}_i) =$ `true` iff $\mathsf{ctr}(\mathcal{Q}_i) = \mathsf{ctr}(\mathcal{Q}_{i-1}) + 1$. In the new game $\mathsf{Adv}_{\kappa}^{\mathsf{B}_\Sigma}(\mathcal{A})$, the predicate $\mathbf{E}(\mathcal{Q})$ is set if and only if $\mathsf{ctr}(\mathcal{Q}_q) > \kappa$.

Proposition 4. *For any* non-adaptive *adversary* $\mathsf{Adv}_{\kappa}^{\mathsf{B}_\Sigma}(\mathcal{A}) \leq \binom{q}{k+1} \left(\frac{B_\Sigma}{q}\right)^{k+1}$.

Note that for $\kappa = 0$ we retrieve the result of the preceding section given that $\binom{q}{1} \left(\frac{B_\Sigma}{q}\right)^{1} = B_\Sigma$ and it might be tempting to think that for larger κ adaptivity can be argued away. This however is not the case, an adaptive adversary does have an increased advantage playing $\mathsf{Adv}_{\kappa}^{\mathsf{B}_\Sigma,\mathbf{B}'}$ when compared to a non-adaptive one. Nonetheless, we conjecture that the bound just derived is sufficiently loose to apply to adaptive adversaries as well.

Conjecture 1. For any *adaptive* adversary $\mathsf{Adv}_{\kappa}^{\mathsf{B}_\Sigma}(\mathcal{A}) \leq \binom{q}{k+1}\left(\frac{B_\Sigma}{q}\right)^{k+1}$.

Proposition 5. *For any* adaptive *adversary* $\mathsf{Adv}_{\kappa}^{\mathsf{B}_\Sigma}(\mathcal{A}) \leq (B_\Sigma)^{\kappa+1}$.

4 A New Double-Length Compression Function

In this section, we introduce a new compression function (Construction 1, see also Fig. 1) $H\colon \{0,1\}^{3n} \to \{0,1\}^{2n}$ that makes parallel calls to two random functions $f_1, f_2\colon \{0,1\}^{2n} \to \{0,1\}^n$. For notational convenience, we often write the input $W \in \{0,1\}^{3n}$ as $(a,b,c) \in (\{0,1\}^n)^3$ and identify $\{0,1\}^n$ with $\mathbb{F}_{2^n}$.

Construction 1. Let $f_1, f_2\colon \{0,1\}^{2n} \to \{0,1\}^n$ be two distinct and independently sampled PuRFs. Define $H^{f_1,f_2}\colon \{0,1\}^{3n} \to \{0,1\}^{2n}$ to be a single-layer compression function using the preprocessing function $C^{\text{PRE}}\colon \mathbb{F}_{2^n}^3 \to (\mathbb{F}_{2^n}^2)^2$ defined by $C^{\text{PRE}} = (C_1^{\text{PRE}}, C_2^{\text{PRE}})$, where

$$C_1^{\text{PRE}}(a,b,c) = (a,b) \quad \text{and} \quad C_2^{\text{PRE}}(a,b,c) = (c, ac+b)$$

and the postprocessing function $C^{\text{POST}}\colon \mathbb{F}_{2^n}^5 \to \mathbb{F}_{2^n}^2$

$$C^{\text{POST}}(a,b,c,y_1,y_2) = A \cdot \begin{pmatrix} a & c & y_1 & y_2 & y_1y_2 \end{pmatrix}^T \;, \text{ where } A = \begin{pmatrix} \omega_{11} & \omega_{12} & \omega_{13} & \omega_{14} & \omega_{15} \\ \omega_{21} & \omega_{22} & \omega_{23} & \omega_{24} & \omega_{25} \end{pmatrix}$$

is a matrix (over $\mathbb{F}_{2^n}$) satisfying certain non-degeneracy conditions (see Table 1).

Design rationale. In the security proofs, we abstract as best as we can the properties required of C^{PRE} and C^{POST}. In practice, we recommend using the matrix (cf. Table 1) $A = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$. Note that in the context of iterating the compression function, one needs to specify which input blocks represent the message block and which ones represent the state or chaining variable. Our security results are independent of this choice. The choice may, however, significantly affect the efficiency of the design.

Incidence-Based Preprocessing. For a single-layer construction, the preprocessing function C^{PRE} fully determines the relationship between the queries made to the primitive on one hand, and the compression function evaluations this enables on the other. Our search is therefore for a preprocessing function C^{PRE} such that yield(q) does not grow too fast as a function of q. In particular, we are interested in whether we can find a C^{PRE} that has good behaviour for $q < 2^{2n}$ as well. It turns out, we can do well by exploiting results from incidence geometry. We note the following theorem that is a finite field version of a theorem of Szemerédi and Trotter over the reals (see, e.g. [16] for an elementary proof).

Theorem 2. *Let $\mathbb{F}$ be a finite field and P (resp. L) be a set of points (resp. lines) in $\mathbb{F}^2$. Let $I(P,L) = \{(p,\ell) \mid (p,\ell) \in P \times L$ and $p \in \ell\}$. Then*

$$|I(P,L)| \leq \min\left(|P||L|^{1/2} + |L|, |L||P|^{1/2} + |P|\right) .$$

Let $(a,b),(c,d) \in \mathbb{F}_{2^n}^2$ denote the query pairs made to f_1 and f_2, respectively. We call a query pair (a,b)–(c,d) *compatible* if and only if $((a,b),(c,d))$ is in the image of C^{PRE}. In addition, a query (a,b) is called (c,d)-compatible or vice versa if the pair (a,b)–(c,d) is compatible. For the preprocessing function C^{PRE} from Construction 1, a pair (a,b)–(c,d) is compatible if and only if $d = ac + b$ is satisfied. Finally, a preprocessing function C^{PRE} satisfies the completion property if and only if (i) (a,b) and c (ii) (c,d) and a uniquely determines a compatible query pair (a,b)–(c,d) for any $a,b,c,d \in \mathbb{F}_{2^n}$.

Proposition 6. *The preprocessing function C^{PRE} from Construction 1 has the completion property and* $\text{yield}(q) \leq q^{3/2} + q$.

Proof. We remark that the completion property can be algebraically verified. To determine the yield, we interpret the (a,b) as the line $y = ax + b$ in $\mathbb{F}_{2^n}^2$ and (c,d) as a point in $\mathbb{F}_{2^n}^2$. This renders bounding the yield an immediate consequence of Theorem 2. To finish the proof, note that the sets $\mathcal{Q}[1]$ and $\mathcal{Q}[2]$ correspond to the lines L and the points P, respectively, and $|I(\mathcal{Q}[2],\mathcal{Q}[1])|$ counts exactly the number of compression function inputs whose mapping can be completely determined by the given queries. Specifying $|\mathcal{Q}[1]| = |\mathcal{Q}[2]| = q$ yields the proposition statement. □

Non-linear Matrix-Style Postprocessing. Our postprocessing is clearly inspired by the use of $\mathbb{F}_{2^n}$-matrices by Rogaway and Steinberger [10], but with the crucial difference that we add the non-linear term $y_1 y_2$. Omitting this non-linear term is fatal for security. For the fully-linear version an adaptive adversary can force its evaluated digest to lie (uniformly) on a *prespecified* set of size 2^n. In contrast, for our construction, the adversary's control is significantly reduced.

Security Claims. We state our security claims for collision and (everywhere) preimage resistance in Theorems 3 and 4, respectively. A sketch of our collision resistance proof is given in Section 5. We refer to the full version for proofs of Theorem 4, Corollaries 1 and 2.

Theorem 3. *Let H^{f_1,f_2} be a single-layer compression function defined by C^{POST} given in Construction 1 where $C^{\text{PRE}}\colon \mathbb{F}_{2^n}^3 \to (\mathbb{F}_{2^n}^2)^2$ is any function that satisfies the completion property. Let $k,\mu,\gamma > 0$ and $\lambda \geq 3$ be integers and let $\kappa = k\lambda + \mu$. Then*

$$\mathsf{Adv}_H^{\mathsf{coll}}(q) \leq \frac{\kappa Y}{2^n} + \frac{q(\gamma^2+1)}{2^{n-1}} + \frac{\binom{q}{\gamma}}{2^{(\gamma-1)n-1}} + 2^{2n}\left(\frac{Y}{2^n}\right)^{k+1} + \frac{\binom{q}{\mu}}{2^{(\mu-1)n-1}} + \frac{\binom{q}{\lambda}}{2^{(\lambda-2)n-1}} .$$

Corollary 1. *Let H^{f_1,f_2} be the compression function given in Construction 1. For every $\delta > 0$ and $q = 2^{2n(1-\delta)/3}$, one has $\mathsf{Adv}_H^{\mathsf{coll}}(q) = o(1)$ as $n \to \infty$.*

Theorem 4. *Let H^{f_1,f_2} be a single-layer compression function defined by C^{POST} given in Construction 1 and an arbitrary C^{PRE} that satisfies the completion property. For any integer $\kappa > 1$, one has*

$$\mathsf{Adv}_H^{\mathsf{epre}}(q) \le 2^{n+1}\binom{q}{\kappa}\left(\frac{1}{2^{n-1}}\right)^{\kappa} + \frac{q}{2^{n-1}} + \frac{\kappa q}{2^{n-1}} \ .$$

Corollary 2. *Let H^{f_1,f_2} be the compression function given in Construction 1. Then for all $\delta > 0$ and $q = 2^{n(1-\delta)}$, it holds that $\mathsf{Adv}_H^{\mathsf{epre}}(q) = o(1)$ as $n \to \infty$.*

5 Proof of Collision Resistance (Theorem 3)

5.1 Overall Strategy

Let $\mathcal{A}$ be a collision-finding adversary making at most q queries to each of the public random functions f_1 and f_2 (without loss of generality, we assume that the adversary makes exactly q queries to both). Our goal is to bound $\mathsf{Adv}_H^{\mathsf{coll}}(\mathcal{A})$, in particular $\Pr[\mathsf{coll}(\mathcal{Q})]$, where $\mathcal{Q}$ is adaptively generated by $\mathcal{A}$. We slightly abuse notation and use $\mathcal{Q}$ (and derived symbols such as $\mathcal{Q}_i$) interchangeably as a random variable (when it is the direct result of playing the collision game), or as a dummy variable (e.g. when we want to quantify over all possible instantiations), where the context makes the precise meaning clear. In all cases we can use the global parameter q for the number of f_1 and f_2 queries and $Y = \text{yield}(q)$.

To bound the probability of an adversary finding a collision, we first look at the probability that any specific query completes the collision: fix i and consider the event $\mathsf{coll}(\mathcal{Q}_i) \wedge \neg\mathsf{coll}(\mathcal{Q}_{i-1})$. Here we call query i *fresh* and we say it *causes* a collision. For concreteness, suppose the ith query is an f_1-query (a, b) (the case for an f_2-query (c, d) is analogous), then the first observation is that it adds a new point to the yield set for every (a, b)-compatible pair (c, d) that was already in $\mathcal{Q}_{i-1}$. Now the ith query can cause a collision in two different ways:

Case I Two compatible and colliding pairs (a, b)–(c, d) and (a', b')–(c', d') are formed with the triple $\{(a', b'), (c, d), (c', d')\} \subseteq \mathcal{Q}_{i-1}$ (where $(a, b) \neq (a', b')$).
Case II Two distinct compatible and colliding pairs exist with $(a, b) = (a', b')$ and $\{(c, d), (c', d')\} \subseteq \mathcal{Q}_{i-1}$, where $(c, d) \neq (c', d')$.

We associate the events $\mathsf{coll}_I(\mathcal{Q})$ and $\mathsf{coll}_{II}(\mathcal{Q})$ with these two cases; it follows that $\mathsf{coll}(\mathcal{Q}) \equiv (\mathsf{coll}_I(\mathcal{Q}) \vee \mathsf{coll}_{II}(\mathcal{Q}))$. The probability of finding a collision at the ith step depends strongly on the number of compatible queries already in $\mathcal{Q}_{i-1}$; we denote this number by (random variable) n_i. While we know (by design) that $\sum_{i=1}^{2q} n_i \le \text{yield}(q)$, a straightforward union bound fails to take this into account properly: Because potentially $n_i \approx q$, naive bounding of $\sum_{i=1}^{2q} n_i$ would be quadratic in q (which is typically much larger than yield(q)). Dealing with this in case of non-adaptive adversaries is straightforward (as such an adversary needs to commit to the n_i values in advance), but requires a more careful treatment in the case of adaptive adversaries. To bound the probability of $\mathsf{coll}_I(\mathcal{Q})$, we additionally condition on not having too many collinear output points. For an integer $\kappa > 0$, $\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})$ is set if and only if $\mathcal{Q}$ leads to more than κ collinear output points in $\mathbb{F}_{2^n}^2$. The reason for collinearity will become evident shortly.

An Overview of the Proof. We start with the observation, for any $\mathcal{Q}$, that

$$\mathsf{coll}(\mathcal{Q}) \;\equiv\; (\mathsf{coll}_I(\mathcal{Q}) \vee \mathsf{coll}_{II}(\mathcal{Q})) \equiv (\mathsf{coll}_I(\mathcal{Q}) \wedge \neg\mathsf{coll}_{II}(\mathcal{Q})) \vee \mathsf{coll}_{II}(\mathcal{Q}) \ ,$$

where the expression $(\mathsf{coll}_I(\mathcal{Q}) \wedge \neg\mathsf{coll}_{II}(\mathcal{Q}))$ is equivalent to

$$\big(\mathsf{coll}_I(\mathcal{Q}) \wedge \neg\mathsf{coll}_{II}(\mathcal{Q}) \wedge \neg\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})\big) \vee \big(\mathsf{coll}_I(\mathcal{Q}) \wedge \neg\mathsf{coll}_{II}(\mathcal{Q}) \wedge \mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})\big) \ .$$

Using the trivial implications for the above statements, we reach

$$\mathsf{coll}(\mathcal{Q}) \Rightarrow \underbrace{\big(\mathsf{coll}_I(\mathcal{Q}) \wedge \neg\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})\big)}_{\mathbf{E_1}} \vee \underbrace{\big(\neg\mathsf{coll}_{II}(\mathcal{Q}) \wedge \mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})\big)}_{\mathbf{E_2}} \vee \underbrace{\mathsf{coll}_{II}(\mathcal{Q})}_{\mathbf{E_3}} . \quad (1)$$

The idea of our proof is to find separate upper bounds for the probability of the events $\mathbf{E_i}$ for $i = 1, 2, 3$ and then use the union bound the finalize the proof in Corollary 3 (i.e. $\sum_{i=1}^{3} \Pr[\mathbf{E_i}]$ provides the overall upper bound). An upper bound for $\Pr[\mathbf{E_1}]$ is given in Lemma 7 (corresponding to the term $\kappa Y/2^n$ in Theorem 3). An upper bound for $\Pr[\mathbf{E_3}]$ is established in Lemma 8, which corresponds to the term $q\gamma^2/2^{n-1} + q^\gamma/2^{(\gamma-1)n-1}$ from Theorem 3. Finally, we explain where the bounds for $\Pr[\mathbf{E_2}]$ (i.e. the remaining terms from Theorem 3) come from. We use Proposition 9 to establish an implication that leads to an upper bound for $\Pr[\mathbf{E_2}]$. Moreover, several auxiliary events, which are defined and investigated in Sections 5.2 and 5.4, are required to finalize the bound $\Pr[\mathbf{E_2}]$: The upper bound for the auxiliary events are given in Lemmas 3, 4, 5 and 9.

On the Matrix A Used in C^{POST}. In the following, we consider a general matrix A (see Construction 1) over $\mathbb{F}_{2^n}$ for the proof of Theorem 3. The conditions on the entries of the matrix A required throughout the paper, as well as where they are used, are provided in Table 1. (Note that the probability that a randomly selected matrix A satisfies our criterion is close to one.)

Output Lines. By assumption, we know that an f_1-query (a, b) can only complete a collision using an already present compatible f_2-query (c, d). Let (a, b) be an f_1-query and let (c, d) be a preceding (a, b)-compatible f_2-query with $y_2 = f_2(c, d)$. The output $C^{\text{POST}}(a, b, c, y_1, y_2)$ of the compression function on input (a, b, c) then lies on the line (in $\mathbb{F}_{2^n}^2$)

$$\mathcal{L}_{1:c,d,y_2;a} \colon \left\{ \underbrace{\begin{pmatrix} a\omega_{11} + c\omega_{12} + y_2\omega_{14} \\ a\omega_{21} + c\omega_{22} + y_2\omega_{24} \end{pmatrix}}_{\text{offset}} + y_1 \underbrace{\begin{pmatrix} \omega_{13} + y_2\omega_{15} \\ \omega_{23} + y_2\omega_{25} \end{pmatrix}}_{\text{slope}} \mid y_1 \in \mathbb{F}_{2^n} \right\} \ , \quad (2)$$

where we get the actual output point for (a, b, c) by setting $y_1 = f_1(a, b)$. The randomness of f_1 results in a random point on $\mathcal{L}_{1:c,d,y_2;a}$. Note that the line cannot be degenerate (see condition **C**1 in Table 1), i.e. it has nonzero slope.

The Condition	Where used	Reference
	(Theorem 3)	(Section 5)
C1: $\omega_{13}\omega_{25} \neq \omega_{15}\omega_{23}$	(**S**) Non-degeneracy of $\mathcal{L}_{1:}$-lines (**N**) Non-parallel $\mathcal{P}_{1:}$-partitions	(2) Lemma 4
C2: $\omega_{14}\omega_{25} \neq \omega_{15}\omega_{24}$	(**S**) Non-degeneracy of $\mathcal{L}_{2:}$-lines (**N**) Non-parallel $\mathcal{P}_{2:}$-partitions	(3) Lemma 4
C3: $\omega_{11} \neq 0 \wedge \omega_{21} \neq 0$	(**N**) Non-degeneracy of $\mathcal{P}_{1:}$-partitions	Lemma 3
C4: $\omega_{12} \neq 0 \wedge \omega_{22} \neq 0$	(**N**) Non-degeneracy of $\mathcal{P}_{2:}$-partitions	Lemma 3
C5: $\omega_{15} \neq 0 \wedge \omega_{25} \neq 0$	(**N**) Nonlinearity of C^{POST}	Construction 1

Table 1. Quick recap of the properties of the entries of A (see Construction 1) used in the proof of Theorem 3. (**N**) denotes that the condition is necessary, whereas (**S**) denotes it is sufficient.

Similarly, let (c, d) be an f_2-query and let (a, b) be a preceding (c, d)-compatible f_1-query. The output of the compression function on (a, b, c) lies on the line:

$$\mathcal{L}_{2:a,b,y_1;c}\colon \left\{ \begin{pmatrix} a\omega_{11} + c\omega_{12} + y_1\omega_{13} \\ a\omega_{21} + c\omega_{22} + y_1\omega_{23} \end{pmatrix} + y_2 \begin{pmatrix} \omega_{14} + y_1\omega_{15} \\ \omega_{24} + y_1\omega_{25} \end{pmatrix} \mid y_2 \in \mathbb{F}_{2^n} \right\} . \quad (3)$$

This time the output point is obtained by setting $y_2 = f_2(c, d)$. Again, the randomness of f_2 results in a random point on $\mathcal{L}_{2:a,b,y_1;c}$. We note that this time non-degeneracy follows from $\omega_{14}\omega_{25} - \omega_{24}\omega_{15} \neq 0$ (see condition **C**2 in Table 1). Now it is easy to see why we do not want too many collinear points: It would ease the collision-finding considerably due to the above output lines.

5.2 Partitions, Bunches and Some Auxiliary Events

Partitions and Bunches. Suppose that an f_2-query (c, d) results in $y_2 = f_2(c, d)$. By the completion property, we obtain, for each $a \in \mathbb{F}_{2^n}$, a unique b such that (a, b) is (c, d)-compatible. Now we recall that if we query $f_1(a, b)$, the resulting yield point lies on the line $\mathcal{L}_{1:c,d,y_2;a}$. From Equation (2) of $\mathcal{L}_{1:c,d,y_2;a}$, it follows that the slope of these lines is fixed (because (c, d) and y_2 are fixed) and independent of a; hence by ranging over all possible $a \in \mathbb{F}_{2^n}$ we achieve a set of (parallel) lines. This is what we call a partition (partitions due to an f_1-query is defined analogously): $\mathcal{P}_{1:c,d,y_2} = \{\mathcal{L}_{1:c,d,y_2;a} \mid a \in \mathbb{F}_{2^n}\}$. The opposite notion to a partition is a bunch: For all preceding and (a, b)-compatible $(c_j, d_j) \in \mathcal{Q}$, for some integer $j \geq 1$, the bunch of interest is the collection of lines (for $y_{2:j} = f_2(c_j, d_j)$)

$$\mathcal{B}_{1:(a,b)}(\mathcal{Q}) = \left\{ \mathcal{L}_{1:c_j,d_j,y_{2:j};a} \mid (c_j, d_j, y_{2:j}) \in \mathcal{Q} \wedge (c_j, d_j) \text{ compatible with } (a, b) \right\} .$$

(We also write $\mathcal{B}_{1:i}$ if the ith query is an f_1-query (a, b).) The answer $y_1 = f_1(a, b)$ specifies a point on each of these lines to be added to the yield set; we refer to this as *realizing* the bunch. For the record, $\mathcal{B}_{2:(c,d)}(\mathcal{Q})$ is defined analogously.

Degenerate Partitions. We have seen that a partition contains a set of parallel lines. If different choices of a lead to different lines, the lines compatible to

(c, d) necessarily partition the output plane (justifying our terminology). It is possible however that regardless of the a values, we end up with identical lines (though with a different parametrization). In such a case, a partition collapses to a single line and we speak of a degenerate partition. A degenerate partition causes problems in our proof, because it allows an adversary to create many collinear points (by ranging over a). Let $\mathsf{bad}_{\mathsf{dp}}(\mathcal{Q})$ denote the event that $\mathcal{Q}$ gives rise to a degenerate partition (either via different a or c values).

Lemma 3. *Let $\mathcal{Q}$ be generated adaptively, then* $\Pr[\mathsf{bad}_{\mathsf{dp}}(\mathcal{Q})] \leq q/2^{n-1}$, *and if* $\omega_{11}\omega_{23} \neq \omega_{13}\omega_{21}, \omega_{12}\omega_{24} \neq \omega_{14}\omega_{22}, \omega_{11}\omega_{25} = \omega_{15}\omega_{21}$ *and* $\omega_{12}\omega_{25} = \omega_{15}\omega_{22}$, *then* $\Pr[\mathsf{bad}_{\mathsf{dp}}(\mathcal{Q})] = 0$.

Parallel Partitions. We now define another bad event, *parallel partitions*, that can potentially help a collision-finding adversary create collinear points. We have seen that, once answered, a single f_2-query (c, d) determines a well-defined slope for the partition $\mathcal{P}_{1:c,d,y_2}$. If two or more distinct partitions (of the same type) have the same slope, we call the partitions parallel. The number of parallel partitions is tightly related to a standard occupancy problem. Consequently, avoiding parallel partitions altogether is not realistic, yet we can put reasonable bounds on too much parallelism occurring. We define $\mathsf{bad}_{\mathsf{pp}[\mu]}(\mathcal{Q})$ to be the event that $\mathcal{Q}$ results in more than μ parallel partitions (of identical type).

Lemma 4. *Let $\mathcal{Q}$ be generated adaptively. Then, for any integer $\mu > 0$,*

$$\Pr\left[\mathsf{bad}_{\mathsf{pp}[\mu]}(\mathcal{Q})\right] \leq \frac{\binom{q}{\mu}}{2^{(\mu-1)n-1}} \ .$$

Local Collinearity. Now we discuss another auxiliary event, *local collinearity*, that is used in our collinearity analysis. Suppose an f_1-query results in $y_1 = f_1(a, b)$. We associate with this query-response pair a point $(a, y_1) \in \mathbb{F}^2_{2^n}$. Let $\mathsf{bad}_{\mathsf{lc}[\lambda]}(\mathcal{Q})$ be the event that there exist at least λ pairs of f_1-queries (a_i, b_i) with distinct a_i values, such that the associated points $(a_i, y_{1:i})$ are collinear or, alternatively, that there exist at least λ pairs of f_2-queries (c_i, d_i) with distinct c_i values, such that the points $(c_i, y_{2:i})$ are collinear.

Lemma 5. *Let $\mathcal{Q}$ be generated adaptively. Then, for any integer $\lambda > 0$,*

$$\Pr\left[\mathsf{bad}_{\mathsf{lc}[\lambda]}(\mathcal{Q})\right] \leq \frac{\binom{q}{\lambda}}{2^{(\lambda-2)n-1}} \ .$$

Target Local Collinearity. For local collinearity, we are interested in any λ associated points being collinear, without worrying about which line they are on. However, in an upcoming case we are only interested in points all lying on a line with a pre-specified slope (the offset of the line is not fixed in advance). Let $\mathsf{bad}_{\mathsf{slc}[\gamma]}(\mathcal{Q})$ be the event that $\mathcal{Q}[1]$ or $\mathcal{Q}[2]$ leads to more than γ associated points collinear with pre-specified, non-vertical slope.

Lemma 6. *Let $\mathcal{Q}$ be generated adaptively. Then, for any integer $\gamma > 0$,*

$$\Pr\left[\mathsf{bad}_{\mathsf{slc}[\gamma]}(\mathcal{Q})\right] \leq \frac{\binom{q}{\gamma}}{2^{(\gamma-1)n-1}} \ .$$

5.3 Bounding Collisions: Focusing on Pr[$\mathbf{E_1}$] and Pr[$\mathbf{E_3}$]

Lemmas 7 and 8 provide an upper bound for $\Pr[\mathbf{E_1}]$ and $\Pr[\mathbf{E_3}]$, respectively.

Lemma 7. *Let i be a positive integer that satisfies $i \leq q$ and Let $\mathcal{Q}_{i-1}$ be arbitrary query list satisfying $\neg\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q}_{i-1})$ (for some positive integer κ). Then*

$$\Pr[\mathbf{E_1}] = \Pr[\mathsf{coll}_I(\mathcal{Q}) \wedge \neg\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})] \leq \frac{\kappa Y}{2^n} \ .$$

Proof (Sketch). We start by noticing that $\Pr[\mathbf{E_1}] \leq \Pr[\mathsf{coll}_I(\mathcal{Q}) | \neg\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})]$. Each of the n_i compatible elements together with the ith query, defines a line such that the random answer to the ith query will determine which point will be added to the yield set. The condition $\neg\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q}_{i-1})$ implies that on each of these lines, there are at most κ previous yield points. Since the underlying primitive is a random function, the answer is fully random and for a given line, one of the previous yield points is hit with probability at most $\kappa/2^n$. A union bound over the n_i lines gives the bound $n_i\kappa/2^n$. To obtain the overall bound, we exploit our refined game $\mathsf{Adv}_{\kappa}^{\mathsf{B}_\Sigma}(\mathcal{A})$ to determine the $\bigvee$ expression. Here we use the above to determine $B_\Sigma = \kappa Y/2^n$ as $\sum_{i=1}^{2q} n_i \leq Y$ (Proposition 3). $\square$

We now bound the probability of finding an instantaneous collision with a fresh query, first given that $\neg\mathsf{bad}_{\mathsf{slc}[\gamma]}(\mathcal{Q}_{i-1})$ holds. Then we finalize our bound for $\Pr[\mathsf{coll}_{II}(\mathcal{Q})]$ using Proposition 2 along with Lemma 6.

Lemma 8. *Let i be a positive integer that satisfies $i \leq q$ and let $\mathcal{Q}$ be generated adaptively. Then, for any integer $\gamma > 0$,*

$$\Pr[\mathbf{E_3}] = \Pr[\mathsf{coll}_{II}(\mathcal{Q})] \leq \frac{q\gamma^2}{2^{n-1}} + \Pr\left[\mathsf{bad}_{\mathsf{slc}[\gamma]}(\mathcal{Q})\right] \ .$$

5.4 Bounding Overall Collinearity: Bounding Pr[$\mathbf{E_2}$]

We now bound $\Pr[\mathbf{E_2}]$. The main technical difficulty is to properly separate the randomness of the f_1- and f_2-queries. In order to do this in the adaptive setting, we use a method that we call *bunching.* For a fixed i, suppose that the ith query is an f_1-query (a,b). Recall that for the f_1-query (a,b), the bunch $\mathcal{B}_{1:i}$ consists of the lines $\mathcal{L}_{1:c_j,d_j,y_{2:j};a}$ for the n_i compatible preceding f_2-queries (c_j,d_j) (with $y_{2:j} = f_2(c_j,d_j)$ for $j = 1,\ldots,n_i$). The answer $y_1 = f_1(a,b)$ adds a single point to the yield set for each compatible f_2-query (c_j,d_j). These n_i new points lie on the lines $\mathcal{L}_{1:c_j,d_j,y_{2:j};a}$, thereby realizing the bunch $\mathcal{B}_{1:i}$. We refer to the set of freshly added points inside a bunch as a *constellation* that we denote by

$$\mathcal{C}_{1:i}(\mathcal{Q}) = \{H^{f_1,f_2}(a,b,c_j) \mid (c_j,d_j) \in \mathcal{Q}_{i-1} \ \wedge \ (c_j,d_j) \text{ compatible with } (a,b)\} \ .$$

In order to determine the maximum collinearity within the yield set, we estimate (i) the probability of too much collinearity occurring within a single constellation (Proposition 8) and (ii) the probability of too many constellations being collinear

(Lemma 9). Here, a set of constellations is collinear if we can choose a point from each constellation in the set such that all chosen yield points are collinear. If we know that at most λ points are collinear within a single constellation, and at most k constellations are collinear, we can conclude that at most $\kappa = k\lambda$ points are collinear overall. This is formalized in the proposition below, taking into account an additional technicality.

Proposition 7. *Let $k, \lambda, \mu > 0$ be fixed integers, $\kappa = k\lambda + \mu$ and let $\mathsf{bad}_{\mathsf{int}[\lambda]}(\mathcal{Q})$ be the event that there exists a constellation having more than λ collinear points. Define $\mathsf{bad}_{\mathsf{ext}[k]}(\mathcal{Q})$ to be the event that there exists a line ℓ passing through more than k constellations whose bunches do not contain ℓ. Then (for arbitrary $\mathcal{Q}$)*

$$\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q}) \Rightarrow \big(\mathsf{bad}_{\mathsf{int}[\lambda]}(\mathcal{Q}) \vee \mathsf{bad}_{\mathsf{ext}[k]}(\mathcal{Q}) \vee \mathsf{bad}_{\mathsf{pp}[\mu]}(\mathcal{Q})\big) \ .$$

Proposition 8 is used to decompose the event $\mathsf{bad}_{\mathsf{int}[\lambda]}(\mathcal{Q})$ into two events.

Proposition 8. *For arbitrary $\mathcal{Q}$, if (integer) $\lambda \geq 3$ then*

$$\big(\neg\mathsf{coll}_{II}(\mathcal{Q}) \wedge \mathsf{bad}_{\mathsf{int}[\lambda]}(\mathcal{Q})\big) \ \Rightarrow \ \big(\mathsf{bad}_{\mathsf{dp}}(\mathcal{Q}) \vee \mathsf{bad}_{\mathsf{lc}[\lambda]}(\mathcal{Q})\big) \ .$$

To bound collinearity between constellations, we first consider collinearity with a given line ℓ in the output plane. We are interested in bounding the probability that at least k constellations are incident to ℓ. For a line ℓ, integer k and query history $\mathcal{Q}$, let $\mathsf{bad}_{\ell-\mathsf{hit}[k]}(\mathcal{Q})$ be the following event: there exist at least k constellations whose corresponding bunches do not contain ℓ that are incident to ℓ. Recall that $\mathsf{bad}_{\mathsf{ext}[k]}(\mathcal{Q})$ is the event that there exists a line ℓ passing through more than k constellations whose bunches do not contain ℓ.

Lemma 9. *Let ℓ be given and let $\mathcal{Q}$ be generated adaptively. Then*

$$\Pr\left[\mathsf{bad}_{\ell-\mathsf{hit}[k]}(\mathcal{Q})\right] \leq \left(\frac{Y}{2^n}\right)^{k+1} \quad \textit{and} \quad \Pr\left[\mathsf{bad}_{\mathsf{ext}[k]}(\mathcal{Q})\right] \leq 2^{2n}\left(\frac{Y}{2^n}\right)^{k+1} \ .$$

Proof (Sketch). Let $\mathsf{ctr}_{\ell-\mathsf{hit}}(\mathcal{Q})$ be the number of constellations that are incident to ℓ, again restricted to those constellations whose corresponding bunch does not contain ℓ. Clearly, the event $\mathsf{bad}_{\ell-\mathsf{hit}[k]}(\mathcal{Q})$ is equivalent to $\mathsf{ctr}_{\ell-\mathsf{hit}}(\mathcal{Q}) \geq k$. Note that for any i, we have $\mathsf{ctr}_{\ell-\mathsf{hit}}(\mathcal{Q}_i) - \mathsf{ctr}_{\ell-\mathsf{hit}}(\mathcal{Q}_{i-1}) \in \{0,1\}$ since constellation i can be counted at most once (namely if it is incident to ℓ). Let $\mathsf{hit}_{\ell-\mathsf{hit}}(i)$ be the event that the bunch $\mathcal{B}_i$ upon realization is incident to ℓ. Suppose that $\ell \notin \mathcal{B}_i$ and that $\mathcal{B}_i$ consists of n_i lines (each containing an output point). Since ℓ intersects each line in a bunch in at most one point, we obtain that $\Pr\left[\mathsf{hit}_{\ell-\mathsf{hit}}(i)\right] \leq n_i/2^n$. Due to yield restrictions, $\sum_{i=1}^{2q} n_i \leq Y$. The lemma statement follows from applying Proposition 4 with $B_\Sigma = Y/2^n$. The statement for $\Pr\left[\mathsf{bad}_{\mathsf{ext}[k]}(\mathcal{Q})\right]$ follows from the union bound over all lines ℓ. □

Proposition 9. *Let k, λ, and μ be positive integers with $\lambda \geq 3$ and $\kappa = k\lambda + \mu$. Then, for arbitrary $\mathcal{Q}$,*

$$\Pr[\neg\mathsf{coll}_{II}(\mathcal{Q}) \wedge \mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})] \leq \Pr[\mathbf{F}(\mathcal{Q})] \ , \textit{where}$$

$$\Pr\left[\mathbf{F}(\mathcal{Q})\right] \leq \Pr[\mathsf{bad}_{\mathsf{ext}[k]}(\mathcal{Q})] + \Pr[\mathsf{bad}_{\mathsf{pp}[\mu]}(\mathcal{Q})] + \Pr[\mathsf{bad}_{\mathsf{lc}[\lambda]}(\mathcal{Q})] + \Pr[\mathsf{bad}_{\mathsf{dp}}(\mathcal{Q})] \ .$$

Finishing the Proof. The following corollary wraps up what we have discussed so far and finishes the proof of Theorem 3 with the help of earlier obtained bounds (Lemmas 3, 4, 5, 7, 8, and 9).

Corollary 3. *Let $\mathcal{Q}$ be generated adaptively, then for $\mathbf{F}(\mathcal{Q})$ given in Prop. 9*

$$\Pr[\mathsf{coll}(\mathcal{Q})] \leq \Pr[\mathsf{coll}_I(\mathcal{Q}) \wedge \neg\mathsf{bad}_{\mathsf{cl}[\kappa]}(\mathcal{Q})] + \Pr[\mathsf{coll}_{II}(\mathcal{Q})] + \Pr[\mathbf{F}(\mathcal{Q})] \ .$$

6 Blockcipher-Based Instantiation

A naïve replacement of the underlying PuRFs in Construction 1 with ideal blockciphers leads to a weaker security due to the availability of the decryption queries (see the full version for the justification). However, adding a layer of "Davies–Meyer" suffices for our purposes. Note that there is no need to change C^{PRE}; the only modification is in C^{POST} (the proofs are given in the full version).

Construction 5. Let $E_1, E_2 \colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be two fixed randomly (and independently) chosen blockciphers. Define a single-layer compression function $H^{E_1,E_2} \colon \{0,1\}^{3n} \to \{0,1\}^{2n}$ by $C^{\text{PRE}} \colon \mathbb{F}_{2^n}^3 \to (\mathbb{F}_{2^n}^2)^2$ from Construction 1 and $C^{\text{POST}} \colon \mathbb{F}_{2^n}^5 \to \mathbb{F}_{2^n}^2$

$$C^{\text{POST}}(a,b,c,y_1,y_2) = A \cdot (a, c, a+y_1, c+y_2, (a+y_1)(c+y_2))^T \ , \text{ where}$$

A is a matrix satisfying certain non-degeneracy conditions.[3]

Theorem 6. *Let H^{E_1,E_2} be given as in Construction 5 where $C^{\text{PRE}} \colon \mathbb{F}_{2^n}^3 \to (\mathbb{F}_{2^n}^2)^2$ is an arbitrary function that satisfies the completion property. Let $k, \mu, \gamma > 0$ and $\lambda \geq 3$ be integers. Then, for $\kappa = k\lambda + \mu$, $\mathsf{Adv}_H^{\mathsf{coll}}(q)$ is upper bounded by*

$$\frac{\kappa Y + 2q\gamma^2 + 4q}{2^n - q} + \frac{4\binom{q}{\gamma}}{(2^n-q)^{(\gamma-1)}} + 2^{2n}\left(\frac{Y}{2^n-q}\right)^{k+1} + \frac{4\binom{q}{\mu}}{(2^n-q)^{(\mu-1)}} + \frac{4\binom{q}{\lambda}}{(2^n-q)^{(\lambda-2)}} \ .$$

Theorem 7. *Let H^{E_1,E_2} be given as in Construction 5 where $C^{\text{PRE}} \colon \mathbb{F}_{2^n}^3 \to (\mathbb{F}_{2^n}^2)^2$ is an arbitrary function that satisfies the completion property and let $\kappa > 1$ be an integer. Then*

$$\mathsf{Adv}_H^{\mathsf{epre}}(q) \leq 2^{n+2}\binom{q}{\kappa}\left(\frac{2}{2^n-q}\right)^{\kappa} + \frac{2q}{2^n-q} + \frac{2\kappa q}{2^n-q} \ .$$

7 Acknowledgments

This work was initiated while the third author was at EPFL. It has been supported in part by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. The first two authors are supported by a grant of the Swiss National Science Foundation, 200021-122162. We thank anonymous reviewers of TCC'12 and Tom Shrimpton for their useful comments.

[3] We note that we require extra conditions on the entries of A to make our proofs work; yet these are minor and can be easily satisfied, e.g. the proposed matrix A.

References

1. Black, J., Rogaway, P., Shrimpton, T.: Black-box analysis of the block-cipher-based hash-function constructions from PGV. In: Yung, M. (ed.) Advances in Cryptography—Crypto'02. LNCS, vol. 2442, pp. 320–335. Springer, Heidelberg (2002)
2. Black, J., Rogaway, P., Shrimpton, T., Stam, M.: An analysis of the block-cipher-based hash functions from PGV. Journal of Cryptology 23(4), 519–545 (2010)
3. Brachtl, B., Coppersmith, D., Hyden, M., Matyas, S., Jr., Meyer, C., Oseas, J., Pilpel, S., Schilling, M.: Data authentication using modification detection codes based on a public one-way encryption function. U.S. Patent No 4,908,861 (1990)
4. Hirose, S.: Some plausible constructions of double-length hash functions. In: Robshaw, M.J. (ed.) FSE'06. LNCS, vol. 4047, pp. 210–225. Springer, Heidelberg (2006)
5. Lai, X., Massey, J.L.: Hash function based on block ciphers. In: Rueppel, R.A. (ed.) Advances in Cryptography—Eurocrypt'92. LNCS, vol. 658, pp. 55–70. Springer, Heidelberg (1992)
6. Lucks, S.: A collision-resistant rate-1 double-block-length hash function. In: Biham, E., Handschuh, H., Lucks, S., Rijmen, V. (eds.) Symmetric Cryptography. No. 07021 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany (2007), `http://drops.dagstuhl.de/opus/volltexte/2007/1017`
7. Maurer, U.M.: Indistinguishability of random systems. In: Knudsen, L. (ed.) Advances in Cryptography—Eurocrypt'02. LNCS, vol. 2332, pp. 110–132. Springer, Heidelberg (2002)
8. Pietrzak, K.: Indistringuishability and Composition of Random Systems. Ph.D. thesis, ETH Zurich (2005)
9. Preneel, B., Govaerts, R., Vandewalle, J.: Hash functions based on block ciphers: A synthetic approach. In: Stinson, D. (ed.) Advances in Cryptography—Crypto'93. LNCS, vol. 773, pp. 368–378. Springer, Heidelberg (1993)
10. Rogaway, P., Steinberger, J.: Constructing cryptographic hash functions from fixed-key blockciphers. In: Wagner [17], pp. 433–450
11. Rogaway, P., Steinberger, J.: Security/efficiency tradeoffs for permutation-based hashing. In: Smart, N.P. (ed.) Advances in Cryptography—Eurocrypt'08. LNCS, vol. 4965, pp. 220–236. Springer, Heidelberg (2008)
12. Stam, M.: Beyond uniformity: Better security/efficiency tradeoffs for compression functions. In: Wagner [17], pp. 397–412
13. Stam, M.: Blockcipher-based hashing revisited. In: Dunkelman, O. (ed.) FSE'09. LNCS, vol. 5665, pp. 67–83. Springer, Heidelberg (2009)
14. Steinberger, J.: The collision intractability of MDC-2 in the ideal-cipher model. In: Naor, M. (ed.) Advances in Cryptography—Eurocrypt'07. LNCS, vol. 4515, pp. 34–51. Springer, Heidelberg (2007)
15. Steinberger, J.P.: Stam's collision resistance conjecture. In: Gilbert, H. (ed.) Advances in Cryptography—Eurocrypt'10. LNCS, vol. 6110, pp. 597–615. Springer, Heidelberg (2010)
16. Tao, T.: The Szemerédi-Trotter theorem and the cell decomposition (2009), `http://terrytao.wordpress.com/2009/06/12/the-szemeredi-trotter-theorem-and-the-cell-decomposition/`
17. Wagner, D. (ed.): Advances in Cryptography—Crypto'08, LNCS, vol. 5157. Springer, Heidelberg (2008)