A new design for low-depth compression functions from length preserving public random functions

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Department of Computer Science
Institute of Theoretical Computer Science

A New Design for Low-Depth Compression Functions from Length Preserving Public Random Functions

Jackey Lui

Master Thesis in Computer Science

July 14th, 2008 – January 14th, 2009

Supervisors: Prof. Dr. Ueli Maurer
Stefano Tessaro

Abstract

A public random function $R : \{0,1\}^m \to \{0,1\}^n$ is a function chosen uniformly at random from the set of all m-bit to n-bit functions, and is accessible by every party, including the adversary. It is a typical model in the design of hash functions. In this paper we investigate compression functions constructed from length-preserving public random functions ($m = n$), and we aim to achieve optimal collision resistance and preimage resistance while maintaining low-depth, i.e. minimizing the number of random functions connected in series.

In particular, we present a $2n$-bit to n-bit compression function consisting of two layers and makes a total of $3t$ calls to the underlying public random functions. For $t \geq 2$, the construction has optimal collision resistance and a preimage resistance of $\Theta(2^{\frac{t+1}{t+2}n})$ queries against non-adaptive adversaries. We also conjecture the same preimage resistance for adaptive adversaries.

Acknowledgements

I would like to thank Stefano Tessaro for his strong support during my the master's project. He gave me a lot of insights on my research which leads to the results I have here. He also gave me helpful feedback and comments during the writing process. He has taught me many things which cannot be learned unless by staying in the field for years. Such knowledge gives me a much better understanding of what I have worked on.

I would also like to thank Martijn Stam for his comments on the technical details of my research, which contribute to the main result of my thesis.

Lastly I would like to thank Professor Ueli Maurer for his lecture on Cryptography, which arouses my interest into working on this field.

This thesis is dedicated to my family and friends, who have given me great support throughout my studies in ETH.

Contents

1 Introduction **3**
- 1.1 Cryptographic Hash Functions . . . 3
- 1.2 Designs of Cryptographic Hash Functions . . . 5
- 1.3 Our Contributions . . . 6
- 1.4 Related Work . . . 6
- 1.5 Notations and Preliminaries . . . 7

2 Public Random Functions **8**
- 2.1 Public Random Primitives . . . 8
  - 2.1.1 Random Primitive Reductions . . . 9
- 2.2 Hash Functions from Public Random Primitives . . . 14
  - 2.2.1 Properties of Hash Functions . . . 14
- 2.3 Existing Constructions . . . 17
  - 2.3.1 Existing Compression Functions . . . 17
  - 2.3.2 Existing Domain Extenders . . . 23

3 The Generalized Benes Construction **25**
- 3.1 The Benes Construction . . . 25
- 3.2 The Generalized Benes Construction . . . 27

4 Collision Resistance of Generalizd Benes Construction **29**
- 4.1 Proof Preparation . . . 30
- 4.2 Bounding Pr[icoll] . . . 33
- 4.3 Bounding Pr[kcoll_{W_1}] . . . 34
- 4.4 Bounding Final-Collision-Finding Advantage . . . 34
- 4.5 Interpretation of Theorem 5 . . . 37

5 Preimage Resistance of Generalized Benes Construction **40**
- 5.1 Tail Inequalities for Random Variables Under Exclusive-Or . . . 42
- 5.2 Preimage Resistance Against Non-Adaptive Adversaries . . . 44
- 5.3 Potential Approaches . . . 47

6 Conclusion **52**

Chapter 1

Introduction

1.1 Cryptographic Hash Functions

A *hash function* is a mapping $h : \{0,1\}^* \to \{0,1\}^n$ which maps an arbitrarily long input to a shorter digest, and is mainly used to enhance searching speed. A typical application of hash functions is database searching, where records are represented by their corresponding hash values. To search in such a database, the search input digest will be computed, saving comparison time by a shortened search key. Moreover, only records with the same hash value as the search input need to be compared using the original input, reducing the number of records to be searched. However, to maximize efficiency one has to use an h which can evenly assign hash values to records in general. Otherwise if too many records are assigned to the same hash value, searching in such a set of records will make little difference from performing a linear search over all records. In the optimal case every record can be represented by a unique hash value without n being too large. Such notion is brought to cryptographic schemes. Given the message hash received in an authenticated manner, the actual message received can be authenticated by computing the hash code based on the message received, then comparing it with the authenticated message hash. If only few messages are mapped to the same hash value, any tampered message will very likely result in a mismatch in the hash code. However, there is a major difference in the problem setting such that hash functions used to quicken searching are not suitable for cyptographic schemes. If a hash function for database algorithms fails to serve its purpose, namely by assigning the same hash value to too many records, efficiency is greatly reduced and nobody is better off. However, in an authentication scheme, having too many messages resulting in the same hash value will allow adversaries to forge messages easily. Hence one expects hash functions suitable for cryptographic schemes to possess some additional properties, and this type of hash functions are called *cryptographic hash functions* $H : \{0,1\}^* \to \{0,1\}^n$.

Ideally, a cryptographic hash function should act like a black-box, producing

unpredictable outputs unless the same input has already been queried previously, i.e. it should behave like a *random oracle*, which returns a uniform and independent n-bit digest for every distinct input. There are many schemes which are proven secure by making use of a random oracle, like the OAEP [5] and PSS [6] used today. However, any practical cryptographic hash function has to be deterministic, no matter it relies on a shared secret key or a fixed key like SHA-1 and MD5. Hence a random oracle is not realizable [7, 11]. Yet, if one wants to restrict adversaries from exploiting the internal structure of some component, such restriction is formulated by assuming the component is a black-box with respect to adversaries. Obviously, analysis cannot be done without defining the behavior of a black-box, and the most natural way of defining its behavior is to treat it as random. Therefore, even though the assumption of the existence of a random oracle, called the *random oracle model*, is strong, it is indeed rooted from a much more reasonable assumption: adversaries are restricted to only mounting generic attacks to certain components of a scheme. As long as adversaries do not violate such restriction, properties of schemes proven under the random oracle model are meaningful. After all, it is better to trust on schemes with security proofs than adhoc schemes without provable security.

Much research has also been done on the replacement of a random oracle by an efficient function on schemes which are proven secure under this model [4]. In order to find substitution candidates, people focus on implementing functions with properties possessed by random oracles. One cannot list all the properties favorable for a cryptographic hash function, but there are some formulated properties needed by many cryptographic schemes:

- *Collision resistant*: Finding two distinct strings s, s' such that $H(s) = H(s')$ is infeasible.
- *k-multicollision resistant*: Finding k distinct strings $s_1, \ldots, s_k$ such that $H(s_1) = \cdots = H(s_k)$ is infeasible.
- *Preimage resistant*: Given string h, it is infeasible to find s such that $H(s) = h$.
- *Second Preimage resistant*: Given s, it is infeasible to find $s' \neq s$ such that $H(s') = H(s)$.

An application of cryptographic hash functions is digital signatures. Suppose the sender wants to send a message. The typical procedure of applying digital signature to the message should be signing the message with the sender's private key. However, since signature schemes may not be able to sign arbitrarily long messages, and can be much slower than applying a hash function first and then signing the digest, the sender will sign the hash value of the message instead of the original message itself. However, the resulting signature does not only match for just one particular message, but all messages that resolves to the same hash value. If the cryptographic hash function used is not second preimage resistant, then the digital signature scheme is vulnerable to forged messages.

Another common application is to verify data-integrity. Suppose a party is sending a file to another party. By applying the same hash function to the file each party has after the transmission, the receiver can ensure the received file is identical to that sent by the sender, namely by first obtaining the hash code from the sender in an authenticated way, and then compare it with the hash code computed from the received file. If the codes do not match, then the received file is guaranteed to be corrupted. Otherwise, both parties will believe that the transmission is successful. If the underlying cryptographic hash function is not collision resistant, the receiver may get a corrupted file without knowing it, since the files from both parties are hashed to the same value.

One cannot expect application builders to be experts in cryptographic hash functions. They tend to use whatever they can find without knowing the explicit security requirements for the underlying hash functions, so SHA-1 and MD5 are still widely used despite of attacks [19, 20]. As the security of many cryptographic schemes rely on specific properties of cryptographic hash functions, and properties needed in the future are unknown, in order to cope with upcoming schemes, it is important to design cryptographic hash functions which can deliever as many properties as possible in the most efficient way while minimizing complexity. The National Institute of Standards and Technology (NIST) addressed this issue, thus organized a cryptographic hash function competition. The resulting algorithm, referred to as SHA-3, will serve as a direct substitution of SHA-2.

1.2 Designs of Cryptographic Hash Functions

Designing a good cryptographic hash function is no easy task, and there are different approaches based on different assumptions. In order to take inputs of varying length, cryptographic hash functions typically adopt an iterative design, with processing components either aligned in a tree form or a single pipeline. Although one can build a cryptographic hash function from scratch, a popular approach is to split the process into two parts: the design of a component function $f : \{0,1\}^m \to \{0,1\}^n$ and the design of a *domain extender* which uses f as a component. f can be either compressing or non-compressing. If f compresses ($m > n$) then it is called a *compressing function*. If f is non-compressing ($m = n$) then it is a *non-compressing function*, usually a block-cipher in practice.

Such design strategy allows new component functions to develop into a cryptographic hash function by using existing domain extenders, or in another perspective, new domain extenders can readily take existing component functions. In order to build a cryptographic hash function with the desired properties, domain extender designs focus on preserving properties of the underlying components. A good example will be the Merkle-Damgård construction [10]. It is well-known for its preservation of collision resistance, which extends the domain of a collision resistant compression function into a hash function achieving approximiately the same collision resistance. On the other hand, component

function designs focus on achieving the desirable properties of a complete cryptographic hash function. If an attack against the compression function is successful, after domain extension the resulting hash function will very likely suffer from the same attack as well. Hence the security of a component function should also be as strong as possible. However, since a component function will be called many times by the domain extender, it should be as efficient as possible too, which seems to impose a constraint on the maximization of security. In fact, it was shown that there is a tradeoff between the efficiency and security of a compression function when some parameters are fixed [18, 16]. Under such constraints, designing a good component function can also be challenging, so researchers naturally subdivide the problem further into building component functions using even smaller components. There are many compression function constructions using non-compressing random systems as components, mostly random functions or ideal ciphers. Although such random primitives do not exist as well, with this approach it is easier for one to design replacement candidates, since a small component is usually simpler to design and analyze comparing to relatively larger components.

1.3 Our Contributions

Under the public random function model, we designed a class of $2n$-bit to n-bit two-layered compression functions H_t, making reference to the Benes construction proposed by Aiello and Venkatesan [1]. Every call of H_t makes one call to each of the $3t$ underlying n-to-n bit random functions. For $t \geq 2$, we proved the collision resistance of H_t to be $\Theta(2^{\frac{n}{2}})$. The preimage resistance against non-adaptive adversaries is $\Theta(2^{\frac{t+1}{t+2}n})$, so for adaptive adversaries in general this is an upper bound. Together with the construction by Shrimpton and Stam [17], the task of finding preimage resistances of both designs resolve into the same mathematical problem, giving support to their corresponding preimage resistance which is conjectured to be $2^{\frac{2}{3}n}$. We also conjecture that the preimage resistance against adaptive adversaries is also $\Theta(2^{\frac{t+1}{t+2}n})$, and suggested approaches which we believe can lead to the final answer.

1.4 Related Work

There is an optimally collision resistant construction $\{0,1\}^{2n} \to \{0,1\}^n$ proposed by Shrimpton and Stam [17] which makes only three calls to f. They claimed the preimage resistance of such construction is $2^{\frac{2}{3}n}$ with little proof, but with an additional call their construction can be both optimally collision and preimage resistant. On the other hand, Rogaway and Steinberger [15] analyzed constructions which make use of random permutations instead of random functions. Under the assumptions "collision uniformity" and "preimage uniformity", they developed a systematic way to examine a family of compression functions, and they claimed the existence of a construction $\{0,1\}^{2n} \to \{0,1\}^n$

which achieves a collision resistance of $2^{\frac{n}{2}}$ and a preimage resistance of $2^{\frac{3}{4}n}$, while making only four calls to the underlying random permutation.

1.5 Notations and Preliminaries

In this section we introduce some notations which will be used throughout the thesis.

For any positive integer k, $\{0,1\}^k$ denotes the set of all bit strings of length k. For strings $x, y \in \{0,1\}^k$, the symbol $\oplus$ denotes the binary bitwise exclusive-or operation (xor), so $x \oplus y$ will denote their bitwise exclusive-or result. $x \| y$ denotes the concatenation of x followed by y. Sometimes we would like to input numbers to functions which only accept strings, so let i_{bin} denote the binary representation of integer i. Let $\mathcal{F}_{m,n}$ be the set of all functions from $\{0,1\}^m$ to $\{0,1\}^n$, and let $f \xleftarrow{\$} S$ denote f being chosen uniformly at random from the set S.

Chapter 2

Public Random Functions

When designing a cryptographic hash function, security statements can only be made with respect to a set of assumptions. Many hash function designs involve the use of an ideal system, and these ideal systems being used are mainly random. Typical random primitives include:

- Random Oracles
- Ideal Ciphers
- Public Random Functions

2.1 Public Random Primitives

A random primitive is public if everyone has direct access to such primitive, including the adversary. In this section three types of random primitives are introduced: the random oracle, the ideal cipher, and the random function. All these primitives will be considered as public.

A random oracle $O : \{0,1\}^* \rightarrow \{0,1\}^n$ is a mapping which returns a random n-bit string uniformly and independently for any new query, but for any previously queried input it will behave just like a function and answer with the same value. Since its domain is infinitely large, it can also act as a source of random bits. According to the paradigm suggested by M. Bellare and P. Rogaway [4], designs using random oracles can yield efficient protocols, namely by first proving a protocol secure using the random oracle, and then replacing the random oracle by an appropriately chosen function.

An ideal cipher $E : \{0,1\}^\kappa \times \{0,1\}^n \rightarrow \{0,1\}^n$ is a function where for every key $k \in \{0,1\}^\kappa$, $E_k = E(k,\cdot)$ is a random n-to-n bit permutation. Everyone can query both E and E^{-1}. This is a popular primitive in the design of compression functions, since in practice they are replaced by block ciphers without the need to design special replacement candidates.

A random function $R : \{0,1\}^m \rightarrow \{0,1\}^n$ is a function drawn uniformly and independently from the set of all functions from $\{0,1\}^m$ to $\{0,1\}^n$. It is

similar to a random oracle, except having a finite domain. There are compression function designs using random functions which preserve the input-lengths ($m = n$), so $R : \{0,1\}^n \to \{0,1\}^n$ itself is non-compressing. In this thesis our compression function construction is based on public random functions.

2.1.1 Random Primitive Reductions

A natural question about random primitives is whether one can replace the other. The answer is yes, but to formally argue if one primitive can be replaced by another we need the notion of indifferentiability originated from Maurer et al. [11]. Let $F = (F^{\text{priv}}, F^{\text{pub}})$ be a system with both a private and public interface. One can imagine the interfaces as two possibly dependent functions or algorithms, where honest parties will interact with the private interface and the public interface is for the adversary. Let distinguisher $\mathcal{D}$ be an algorithm which takes in a system and returns either 0 or 1, i.e. $\mathcal{D}(F^{\text{priv}}, F^{\text{pub}}) = 0$ or 1. Since a system has two interfaces, $\mathcal{D}$ can choose to query both interfaces, but one query can only allow $\mathcal{D}$ to interact with one interface, not both. Note that $\mathcal{D}$ can be computationally unbounded.

Definition 1. *$(F^{\text{priv}}, F^{\text{pub}})$ is ϵ-indifferentiable from $(G^{\text{priv}}, G^{\text{pub}})$, denoted $F \overset{\epsilon}{\sqsubset} G$, if there exists a system S (called a simulator) such that for any distinguisher $\mathcal{D}$ making at most q queries,*

$$\left|\Pr[\mathcal{D}(F^{\text{priv}}, F^{\text{pub}}) = 1] - \Pr[\mathcal{D}(G^{\text{priv}}, S(G^{\text{pub}})) = 1]\right| \leq \epsilon$$

Note that the notion of indifferentiability is asymmetric in general. Given $F \overset{\epsilon}{\sqsubset} G$, showing $G \sqsubset F$ may require a different simulator, thus having a possibly different ϵ. The notion of indifferentiability is transitive though, shown by the following lemma:

Lemma 1. *If $F \overset{\epsilon}{\sqsubset} G$ and $G \overset{\epsilon'}{\sqsubset} H$, then $F \overset{\epsilon+\epsilon'}{\sqsubset} H$.*

Proof. Let S and S' be simulators such that for any distinguisher $\mathcal{D}$ making at most q queries,

$$\left|\Pr[\mathcal{D}(F^{\text{priv}}, F^{\text{pub}}) = 1] - \Pr[\mathcal{D}(G^{\text{priv}}, S(G^{\text{pub}})) = 1]\right| \leq \epsilon$$
$$\left|\Pr[\mathcal{D}(G^{\text{priv}}, G^{\text{pub}}) = 1] - \Pr[\mathcal{D}(H^{\text{priv}}, S'(H^{\text{pub}})) = 1]\right| \leq \epsilon'$$

Consider the expression

$$\left|\Pr[\mathcal{D}(F^{\text{priv}}, F^{\text{pub}}) = 1] - \Pr[\mathcal{D}(H^{\text{priv}}, S'(S(H^{\text{pub}}))) = 1]\right|$$

By triangle inequality $|a - b| \leq |a - c| + |b - c|$ we have

$$\begin{aligned}
&\left|\Pr[\mathcal{D}(F^{\text{priv}}, F^{\text{pub}}) = 1] - \Pr[\mathcal{D}(H^{\text{priv}}, S'(S(H^{\text{pub}}))) = 1]\right| \\
&\leq \left|\Pr[\mathcal{D}(F^{\text{priv}}, F^{\text{pub}}) = 1] - \Pr[\mathcal{D}(G^{\text{priv}}, S(G^{\text{pub}})) = 1]\right| + \\
&\left|\Pr[\mathcal{D}(G^{\text{priv}}, S(G^{\text{pub}})) = 1] - \Pr[\mathcal{D}(H^{\text{priv}}, S'(S(H^{\text{pub}}))) = 1]\right| \\
&\leq \epsilon + \left|\Pr[\mathcal{D}(G^{\text{priv}}, G^{\text{pub}}) = 1] - \Pr[\mathcal{D}(H^{\text{priv}}, S'(H^{\text{pub}})) = 1]\right| \\
&\leq \epsilon + \epsilon'
\end{aligned}$$

□

Let C be a construction, F a random primitive, and consider the system $C(F)$. Since F is public any adversary can access F, so C can only modify the private interface of F, i.e. $C(F) = (C(F^{\text{priv}}), F^{\text{pub}})$. We say G is reducible to F if there exists a construction C such that $(C(F^{\text{priv}}), F^{\text{pub}})$ is indifferentiable from $(G^{\text{priv}}, S(G^{\text{pub}}))$.

Because indifferentiability is transitive, it suffices to show that E is reducible to O, R is reducible to E, and O is reducible to R, then any primitive is reducible to the other two. All constructions represented are from [8] and [9] by Coron et al.

E is Reducible to O

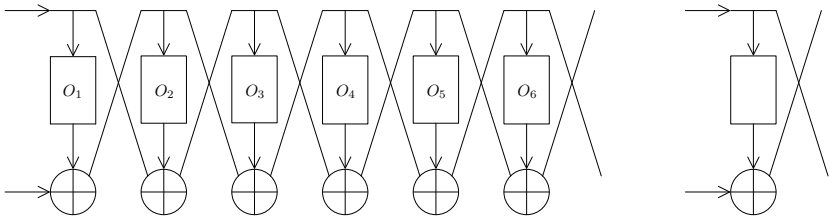

Figure 2.1: The 6-Round Luby-Rackoff Construction (left); A Feistel transformation (right)

Coron et al. presented the 6-round Luby-Rackoff construction (also called the 6-round Feistel network, see Figure 2.1) in [9] which makes $E : \{0,1\}^{\kappa} \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ reducible to O. The formula of a Feistel transformation containing random primitive F is

$$\text{Ft}_F(s_1, s_2) = (s_2, s_1 \oplus F(s_2))$$

Here is the algorithm for the construction:

Algorithm LR$(s_1 \| s_2)$
$y_1 \leftarrow s_1$, $y_2 \leftarrow s_2$
for $i \leftarrow 1$ to 6 **do**
  $(y_1, y_2) \leftarrow \text{Ft}_{O_i}(y_1, y_2)$
end for
return $y_1 \| y_2$

In order to integrate all six random oracles into one, as well as to feed a key to the random oracle, whenever O_i needs to be evaluated on the input x, $O(i_{\text{bin}} \| k \| x)$ is evaluated, where $1 \leq i \leq 6$ and k is the key. According to [9] the 6-round Luby-Rackoff construction is $\left(2^{18} \cdot \frac{q^8}{2^n}\right)$-indifferentiable from E. They also showed that a 5-round Luby-Rackoff construction is insufficient to be indifferentiable from a random permutation, implying the 6-round construction being optimal. Note that the same construction can also be used to prove that E is reducible to R since an infinite domain for the underlying component is not necessary.

O is Reducible to R

In [8], Coron et al. supplied four different constructions which can prove that O is reducible to R, and they are all variants of the Merkle-Damgård construction (See Figure 2.2). Here $f : \{0,1\}^{n+\kappa} \to \{0,1\}^n$ is a public random function, IV is a fixed n-bit string, and all blocks $s_1, \ldots, s_l$ are κ-bit strings unless specified otherwise. It is known that the plain Merkle-Damgård construction has problems as a domain extender, and there are several ways of fixing the problems. These four variants are all based on fixes which makes the Merkle-Damgård construction preserve collision resistance.

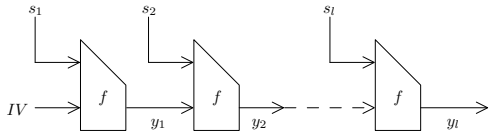

Figure 2.2: The plain Merkle-Damgård Construction

Algorithm MD$(s_1 \| \cdots \| s_l)$
$y_0 \leftarrow IV$
for $i \leftarrow 1$ to l **do**
  $y_i \leftarrow f(y_{i-1}, s_i)$
end for
return y_l

The first construction is called the *Prefix-free Merkle-Damgård Construction*. As the name suggests, a prefix-free encoding of the input is fed to the plain construction. Coron et al. showed that if the underlying component function is a random function, the construction is actually indifferentiable from a random oracle, regardless of any prefix-free encoding used. Figure 2.3 below is an example using a particular type of prefix-free encoding.

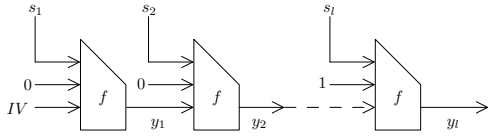

Figure 2.3: The Prefix-free Merkle-Damgård Construction

Algorithm PfMD$(s_1 \| \cdots \| s_l)$
let $g(s)$ be a prefix free encoding of s.
$y \leftarrow$ **MD**$(g(s_1 \| \cdots \| s_l))$
return y

Slightly different from the plain construction, blocks $s_1, \ldots, s_{l-1}$ have size $\kappa - 1$, with the last block s_l padded with 10^r such that $|s_l| + r + 1 = \kappa - 1$.

The second construction is called *The Chop Solution*. Instead of having a prefix-free encoding, bits of the output y_l are truncated, thus the name Chop Solution. Otherwise it is exactly the same as the plain Merkle-Damgård construction. Note that the output string length of the construction is $n - s$, where s is the number of bits chopped. Figure 2.4 is the diagram of the Chop Solution.

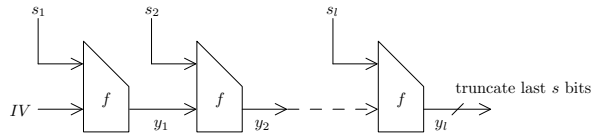

Figure 2.4: The Chop Merkle-Damgård Construction

Algorithm ChopMD$(s_1 \| \cdots \| s_l)$
$y \leftarrow$ **MD**$(s_1 \| \cdots \| s_l)$
return the first $n - s$ bits of y

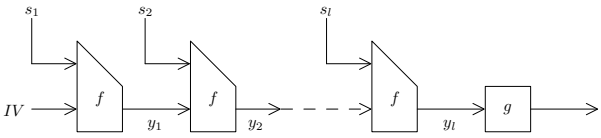

Figure 2.5: The NMAC Construction

Algorithm NMAC$(s_1 \| \cdots \| s_l)$
$y_l \leftarrow$ **MD**$(s_1 \| \cdots \| s_l)$
$y \leftarrow g(y_l)$
return y

The third construction is called the *NMAC* construction (See Figure 2.5). NMAC extends the plain Merkle-Damgård chain by an extra random function $g : \{0,1\}^n \rightarrow \{0,1\}^{n'}$ independent from f.

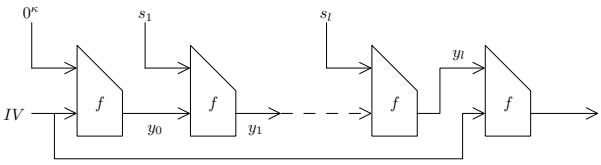

Figure 2.6: The HMAC Construction

Algorithm HMAC$(s_1 \| \cdots \| s_l)$
$s_0 \leftarrow 0^\kappa$

$y_l \leftarrow \mathbf{MD}(s_0 \| s_1 \| \cdots \| s_l)$
if $n < \kappa$ **then**
  $y' \leftarrow y_l \| 0^{\kappa - n}$
else
  $y' \leftarrow y_l|_\kappa$
end if
$y \leftarrow \mathbf{MD}(y')$
return y

The last construction is called the *HMAC* construction, shown in Figure 2.6. IV continues to have size n, but y_l is either padded or truncated to a size of κ, depending on whether $\kappa < n$. Its design is similar to NMAC, but instead of having another random function g, it is replaced by f connected in a special way. Such replacement comes with a tradeoff of having a preliminary phase. The role of $y_0 = f(IV \| 0^\kappa)$ is to prevent the final call of f from using the same initialization vector.

Let l be the maximum length of a query made by the distinguisher $\mathcal{D}$. The following table summarizes the indifferentiability of the four constructions with the corresponding random oracles:

Name	Output size of O	ϵ
Prefix-free MD	n	$2^{-n} l^2 O(q^2)$
Chop MD	n	$2^{-s} l^2 O(q^2)$
NMAC	n'	$2^{-\min(n,n')} l^2 O(q^2)$
HMAC	n	$2^{-\min(n,\kappa)} l^2 O(q^2)$

Table 2.1: Reduction Results of Random Oracles to Random Functions

The proof of all four constructions is by induction, proving a chain of any length is indifferentiable from a random function.

R is Reducible to E

Also the work by Coron et al. [8], they prove that O is reducible to E using the same four constructions described above, namely the underlying random function f can be replaced by the Davies-Meyer compression function (See Figure 2.7), and their corresponding ϵ resembles to the ones in Table 2.1. Consider

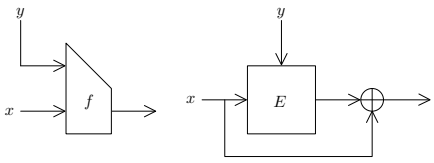

Figure 2.7: The Davies-Meyer Compression Function $E_y(x) \oplus x$

a distinguisher $\mathcal{D}$ making queries with a maximum length of l. If l is shorter than the inputs size of a random function R, then $\mathcal{D}$ can never distinguish a random oracle from R. Therefore the bounds in Table 2.1 also holds for random functions with input sizes at least l, so R is reducible to E.

2.2 Hash Functions from Public Random Primitives

In the last section we see that a random oracle can be replaced by a public random function or an ideal cipher-based construction. As shown in [7, 11], a random oracle is not realizable, therefore all public random primitives are not implementable. However, they still appear in hash function designs because the original assumption is to assume adversaries treating specified components as black-boxes. To model a component being treated as a black-box, we assume it has an output distribution instead of being deterministic. A further assumption is to assume such output distribution being uniform, giving rise to random primitives. Hence the use of public random primitives should not be mistaken as being unrealistic. Given a hash function containing public random primitives, even when they are replaced by real functions, the original security statements will still hold for generic adversaries who do not exploit the internal structures of the replacements.

Because it is reasonable to utilize public random primitives in a design, the ideal goal is to design hash functions which are indifferentiable from a random oracle using public random primitives. In the practical point of view, the goal is to design hash functions such that, as long as adversaries treat its components as black-boxes, the whole construction is no different from a black-box. In fact, such construction exists with nearly optimal security. Designed by Maurer and Tessaro [12], their construction extends the domain of public random functions and is indifferentiable from a public random function up to $\Theta(2^{n(1-\epsilon)})$ queries for any $\epsilon > 0$. However, it is too inefficient to be used in practice. If hash functions which are indifferentiable from random oracles come with a high cost in efficiency, an alternative is to design hash functions with weaker properties, properties which are identifiable from a random oracle. The goal is then to design a hash function with as many properties as possible, but is still efficient to be used by schemes.

2.2.1 Properties of Hash Functions

As more schemes developed, more properties of hash functions are identified. Formally, properties are defined in advantages with respect to adversaries performing certain attacks.

Let $H : M \to \{0,1\}^n$ be a hash function containing r public random primitives $f_1, \ldots, f_r$ with the same domain and output size. If they are public random functions, then they are initialized by being sampled uniformly at random from

$\mathcal{F}_{\alpha,\beta}$ for some α and β, indicated by the expression $f_1, \ldots, f_r \xleftarrow{\$} \mathcal{F}_{\alpha,\beta}$. If they are ideal ciphers, then they do not need to be initialized. Note that M is the domain of H and can be infinitely large. Let $\mathcal{A}$ be an adversary, formulated as an algorithm. Here is a list of an adversary's advantages in the case of public random functions, all defined by Rogaway and Shrimpton [14]. In case of ideal ciphers, drop the expression $f_1, \ldots, f_r \xleftarrow{\$} \mathcal{F}_{\alpha,\beta}$.

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{Coll}}_{H}(\mathcal{A}) &= \Pr\Big[f_1, \ldots, f_r \xleftarrow{\$} \mathcal{F}_{\alpha,\beta}; X, X' \leftarrow \mathcal{A}^{f_1,\ldots,f_r} : \\
&\qquad X \neq X' \text{ and } H^{f_1,\ldots,f_r}(X) = H^{f_1,\ldots,f_r}(X')\Big] \\
\mathbf{Adv}^{\mathrm{Pre}[m]}_{H}(\mathcal{A}) &= \Pr\Big[f_1, \ldots, f_r \xleftarrow{\$} \mathcal{F}_{\alpha,\beta}; X \xleftarrow{\$} \{0,1\}^m; Y \leftarrow H^{f_1,\ldots,f_r}(X); \\
&\qquad X' \leftarrow \mathcal{A}^{f_1,\ldots,f_r}(Y) : H^{f_1,\ldots,f_r}(X') = Y\Big] \\
\mathbf{Adv}^{\mathrm{Sec}[m]}_{H}(\mathcal{A}) &= \Pr\Big[f_1, \ldots, f_r \xleftarrow{\$} \mathcal{F}_{\alpha,\beta}; X \xleftarrow{\$} \{0,1\}^m; X' \leftarrow \mathcal{A}^{f_1,\ldots,f_r}(X) : \\
&\qquad X \neq X' \text{ and } H^{f_1,\ldots,f_r}(X) = H^{f_1,\ldots,f_r}(X')\Big]
\end{aligned}$$

These three advantages, corresponding to *collision resistance* (Coll), *preimage resistance* (Pre), *second preimage resistance* (Sec), are the most common and the most concerned. There are also extensions and variants derived from these advantages:

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{Coll}[k]}_{H}(\mathcal{A}) &= \Pr\Big[f_1, \ldots, f_r \xleftarrow{\$} \mathcal{F}_{\alpha,\beta}; X_1, \ldots, X_k \leftarrow \mathcal{A}^{f_1,\ldots,f_r} : \\
&\qquad X_i \neq X_j \text{ for } i \neq j \text{ and } H^{f_1,\ldots,f_r}(X_1) = \cdots = H^{f_1,\ldots,f_r}(X_k)\Big] \\
\mathbf{Adv}^{\mathrm{ePre}}_{H}(\mathcal{A}) &= \Pr\Big[(Y, S) \leftarrow \mathcal{A}(); f_1, \ldots, f_r \xleftarrow{\$} \mathcal{F}_{\alpha,\beta}; X \leftarrow \mathcal{A}^{f_1,\ldots,f_r}(S) : \\
&\qquad H^{f_1,\ldots,f_r}(X) = Y\Big] \\
\mathbf{Adv}^{\mathrm{eSec}[m]}_{H}(\mathcal{A}) &= \Pr\Big[(X, S) \leftarrow \mathcal{A}(); f_1, \ldots, f_r \xleftarrow{\$} \mathcal{F}_{\alpha,\beta}; X' \leftarrow \mathcal{A}^{f_1,\ldots,f_r}(S) : \\
&\qquad X \neq X' \text{ and } H^{f_1,\ldots,f_r}(X) = H^{f_1,\ldots,f_r}(X')\Big]
\end{aligned}$$

Coll[k] corresponds to *k-way collision resistance*, which is the same as Coll if $k = 2$. The other two definitions are from [14] as well, corresponding to *everywhere-* (second) preimage resistances (ePre,eSec). These are the most general definitions regarding to hash functions. If more is known about the internal structure, more specific definitions can be made.

Adversary Capabilities Before one can say anything about the advantages, the abilities of an adversary must be properly specified. There are two classes of adversaries: *computational* and *information-theoretic*. A computational adversary has bounded computational power and can only run efficient algorithms using an efficient amount of space. In the latter case an information-theoretic

adversary has unbounded computational power. For public random function $R : \{0,1\}^m \to \{0,1\}^n$, $\mathcal{A}$ can make the query (R, x) and will receive the reply $R(x)$. For ideal cipher $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$, $\mathcal{A}$ can either make query $(1, E, k, x)$ or $(0, E, k, x)$, which will be answered by $E(k, x)$ and $E^{-1}(k, x)$ respectively. Since information-theoretic adversaries are computationally unbounded, their ability is only bounded by the number of queries to the underlyimg primitive functions. Information-theoretic adversaries are sometimes being criticized of being too powerful, since the time-space complexity of managing query results and computing the answer in an attack is omitted. Nonetheless, a security statement against information-theoretic adversaries is able to provide lower bounds for the attack costs mounted by computationally bounded adversaries.

Based on how queries are made, adversaries can also be divided into adaptive adversaries or non-adaptive adversaries. An adaptive adversary is allowed to make computations between two queries, thus will be able to adapt based on query results. On the other hand, a non-adaptive adversary must prepare a set of queries beforehand. Once the set of queries is determined, the answers are returned and the adversary can no longer make more any queries.

Based on these advantages as well as the type of adversaries in concern, the resistance of a hash function is a loose concept of how powerful an adversary has to be in order to pose a threat. For example, if $\mathcal{A}$ is an information-theoretic adaptive adversary making $q(n)$ queries, and

$$\mathbf{Adv}_H^{\mathrm{Coll}}(\mathcal{A}) \leq \frac{q^2}{2^n}$$

then H is secure against $\mathcal{A}$ unless q is close to $2^{n/2}$, and we can say that the collision resistance of H is $\Theta(2^{n/2})$. Anyone will be convinced that an adversary with only negligible advantage is not a threat, but the notion is loose regarding non-negligible advantages. Suppose

$$\mathbf{Adv}_H^{\mathrm{Coll}}(\mathcal{A}) = \frac{1}{\log n}$$

The collision-finding advantage of $\mathcal{A}$ still converges to 0, but obviously it is far too slow. On the other hand, if

$$\mathbf{Adv}_H^{\mathrm{Coll}}(\mathcal{A}) = \frac{1}{n^{100}}$$

the collision-finding advantage of $\mathcal{A}$ is still non-negligible, but the advantage converges so fast that H is still secure against $\mathcal{A}$. Therefore the statement "the collision resistance of H is B queries" is just an informal saying, meaning there are adversaries (of a certain class) who can find a collision with "reasonable" probability given B queries.

Out of the eight properties mentioned, some properties have implications on other properties. For example, the collision resistance of any hash function can

never be higher than its second preimage resistance. In terms of advantages defined in this section, for any adversary (of a certain type) $\mathcal{A}$,

$$\max_{\mathcal{A}} \left(\mathbf{Adv}_H^{\mathrm{Sec}[m]}(\mathcal{A}) \right) \leq \max_{\mathcal{A}} \left(\mathbf{Adv}_H^{\mathrm{Coll}}(\mathcal{A}) \right)$$

In this case we shall denote this implication by $\mathbf{Adv}_H^{\mathrm{Coll}} \rightarrow \mathbf{Adv}_H^{\mathrm{Sec}[m]}$. Here is a list of implications (partly quoted from [14]):

1. $\mathbf{Adv}_H^{\mathrm{Coll}[k]} \rightarrow \mathbf{Adv}_H^{\mathrm{Coll}}$ for all $k \geq 2$
2. $\mathbf{Adv}_H^{\mathrm{Coll}} \rightarrow \mathbf{Adv}_H^{\mathrm{Sec}[m]}$
3. $\mathbf{Adv}_H^{\mathrm{Coll}} \rightarrow \mathbf{Adv}_H^{\mathrm{eSec}[m]}$
4. $\mathbf{Adv}_H^{\mathrm{ePre}} \rightarrow \mathbf{Adv}_H^{\mathrm{Pre}[m]}$
5. $\mathbf{Adv}_H^{\mathrm{eSec}[m]} \rightarrow \mathbf{Adv}_H^{\mathrm{Sec}[m]}$

2.3 Existing Constructions

A compression function and a domain extender can combine together into a complete hash function. In this section we will introduce existing compression functions constructed from random functions or ideal ciphers, as well as domain extenders which preserves properties possessed by compression functions.

2.3.1 Existing Compression Functions

All constructions we are going to introduce here share the same adversarial model: information-theoretic adversaries making $q(n)$ queries to the underlying random functions/ideal ciphers. Most of them are adaptive, so assume any adversary to be adaptive unless specified. Moreover, analysis conducted on these constructions all contain information about their collision and preimage resistances, thus will be what we mainly compare and discuss here.

It is obvious that more primitive calls can improve security at a cost in efficiency, so compression functions constructed from public random primitives are usually classified by the number of primitive calls. Another criterion for classification is the number of layers: the maximum number of random primitives connected in series. The number of layers has an effect on efficiency as well because a longer pipeline requires more time to finish computation.

Construction by Shrimpton and Stam

Figure 2.8 shows the compression function designed by Shrimpton and Stam [17]. They did not give their design a name, so we shall denote their construction by

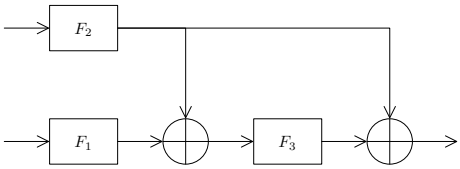

Figure 2.8: Construction by Shrimpton and Stam [17]

SSt : $\{0,1\}^{2n} \to \{0,1\}^n$. F_1, F_2 and F_3 are n-bit to n-bit random functions. In algebraic form, the construction is

$$\mathbf{SSt}(s_1 \| s_2) = F_3(F_1(s_1) \oplus F_2(s_2)) \oplus F_2(s_2)$$

Every query of **SSt** thus uses three calls. F_1 and F_3 are connected in series, so this is a two-layered construction. According to [17] **SSt** has nearly optimal collision resistance and they conjectured the preimage resistance to be $\Theta(2^{2/3n})$.

One remark for their construction is the achievement of optimal preimage resistance by attaching an extra random function after F_3 and becoming a three-layered construction, i.e.

$$F_4(F_3(F_1(s_1) \oplus F_2(s_2)) \oplus F_2(s_2))$$

The argument is simply by reduction. If a preimage of the enhanced construction is found, then that preimage is a preimage of F_4. Since F_4 has optimal preimage resistance, the enhanced construction also has preimage resistance. Moreover, collision resistance is preserved, because F_4 has optimal collision resistance too. Denote this enhanced construction by **eSSt**.

Another way of modifying the construction is to replace the random functions by ideal ciphers. In the same paper Shrimpton and Stam proved that their construction is still optimally collision resistant when F_1 and F_2 are replaced by fixed-key ideal ciphers in Davies-Meyer mode (See Figure 2.7).

Constructions by Stam

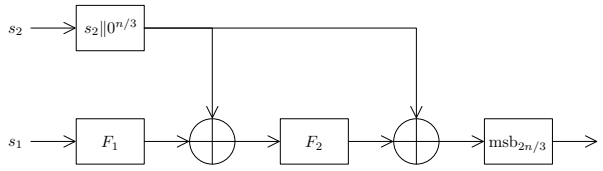

Figure 2.9: A 2-Call Compression Function by Stam [18]

In [18] Stam presented a miniature of **SSt**, with an input size of $5n/3$ and output size of $2n/3$ and makes only two random function calls. This is also a two-layered construction since $\mathrm{msb}_{2n/3}$ is a deterministic function chopping $n/3$

bits away. One can see that it is very similar to **SSt** (Figure 2.9). It also has almost optimal collision resistance (in this case, $2^{n/3}$ due to smaller output size), and the same holds when F_1 is replaced by a fixed-key ideal cipher in Davies-Meyer mode. In fact, he has shown an actual bound on the collision-finding advantage:

$$\mathbf{Adv}_H^{\mathrm{Coll}}(\mathcal{A}) \leq \frac{q^2}{2^{n+1}} + 2^{n/3}\left(\frac{q}{2^{n/3}}\right)^n + \frac{q(q-1)n^2}{2^{2n/3}}$$

This construction can easily extend to an input size of $2n$ and output size of n, namely by forwarding the remaining bits untouched. Call this extended compression function $\mathbf{eSt}_{2n/3}$.

Algorithm $\mathbf{eSt}_{2n/3}(s_1 \| s_2)$
Split s_2 into $a \| b$, where $b \in \{0,1\}^{n/3}$
$y' \leftarrow F_2(F_1(s_1) \oplus s_2) \oplus s_2$
$y \leftarrow \mathrm{msb}_{2n/3}(y')$
return $y \| b$

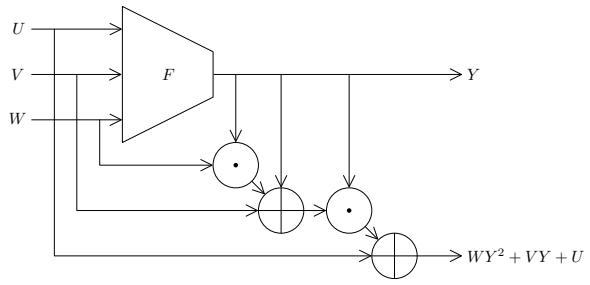

Figure 2.10: A Single Call Double-Length Compression Function by Stam [18]

There is another construction in [18] by Stam, a $3n$-bit to $2n$-bit compression function using a $3n$-bit to $2n$-bit random function. It makes only one call, thus consists of one layer. In Figure 2.10, input strings U, V, W are treated as elements in the finite field $\mathbb{F}_{2^n}$, so the symbols $\oplus$ and $\odot$ refer to addition and multiplication in $\mathbb{F}_{2^n}$ respectively, and the output string of $H(U \| V \| W)$ is $Y \| (WY^2 + VY + U)$. Denote this construction by **StDL**.

Algorithm StDL$(U \| V \| W)$
$Y \leftarrow F(U \| V \| W)$
return $Y \| (WY^2 + VY + U)$

Its security is only shown against non-adaptive adversaries, but the statement is a positive one: For any non-adaptive adversary $\mathcal{A}$ making at most q queries,

$$\mathbf{Adv}_H^{\mathrm{Coll}}(\mathcal{A}) \leq \frac{q(q-1)}{2^{2n}}$$

Hence its collision resistance against non-adaptive adversaries is $\Theta(2^n)$, which is again optimal.

Results by Rogaway and Steinberger

In [15], Rogaway and Steinberger proposed a family of compression functions constructed from fixed-key ideal ciphers. Since the keys are fixed they are equivalent to random permutations. Every construction $\mathbf{LP}^A_{mkn} : \{0,1\}^m \to \{0,1\}^n$ they propose, which makes k calls to the underlying ideal cipher(s), is expressed in the form of a $(k+r) \times (k+m)$ matrix A over $\mathbb{F}_{2^n}$. Let π_i be a random permutation and a_i be the ith row of A. To evaluate $\mathbf{LP}^A_{mkn}$, they provided the following algorithm:

Algorithm $\mathbf{LP}^A_{mkn}(s_1 \| \cdots \| s_m)$
for $i \leftarrow 1$ to k **do**
  $x_i \leftarrow a_i \cdot (s_1, \ldots, s_m, y_1, \ldots, y_{i-1})$
  $y_i \leftarrow \pi_i(x_i)$
end for
for $i \leftarrow 1$ to n **do**
  $w_i \leftarrow a_{k+i} \cdot (s_1, \ldots, s_m, y_1, \ldots, y_k)$
end for
return $w_1 \| \cdots \| w_r$

They explained that the analysis process is automated, and their results presented are all summarized. The only solid construction they showed is $\mathbf{LP}^A_{231}$ where

$$A = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 2 \end{bmatrix}$$

In the analysis process, where m, k, n are fixed, a large number of matrices are tested, and the performance of the best matrix is recorded. Table 2.2 is a summary of their results.

Scheme	Collision Resistance	Preimage Resistance
$\mathbf{LP}_{231}, \mathbf{lp}_{231}$	$2^{0.5n}$	$2^{0.67n}$
$\mathbf{LP}_{241}, \mathbf{lp}_{241}$	$2^{0.5n}$	$2^{0.75n}$
$\mathbf{LP}_{352}, \mathbf{lp}_{352}$	$2^{0.55n}$	$2^{0.80n}$
$\mathbf{LP}_{362}, \mathbf{lp}_{362}$	$2^{0.63n}$	$2^{0.80n}$
$\mathbf{LP}^{\mathbf{SS}}$	$2^{0.50n}$	$2^{0.50n}$
$\mathbf{lp}^{\mathbf{SS}}$	2^{0n}	2^{0n}

Table 2.2: Summary of Automated Analysis by Rogaway and Steinberger

$\mathbf{lp}_{mkn}$ is a scheme similar to $\mathbf{LP}_{mkn}$, but instead of using k fixed-key ideal ciphers, a single fixed-key ideal cipher is used throughout. $\mathbf{LP}^{\mathbf{SS}}$ is the scheme in Figure 2.8 with F_1 and F_2 replaced by two fixed-key ideal ciphers in Davies-Meyer mode. Optimal collision resistance is expected, as proved by Shrimpton

and Stam [17], but its preimage resistance is only approximately $2^{n/2}$. This does not disprove their conjecture that **SSt** has a preimage resistance of $2^{2n/3}$ though, since they make this security statement with respect to F_1 and F_2 being random functions. Finally $\mathbf{lp}^{\mathbf{SS}}$ is the resulting scheme by having F_1 and F_2 replaced by the same random permutation in Davies-Meyer mode, and the result shows that it is a failure.

Let **GBe** be our Generalized Benes construction with $t \geq 2$. Complete details are included in the later chapters. Together with the results of all constructions mentioned in this section, here is a comparison table of constructions. Table 2.3 shows that constructions from ideal ciphers have more layers in gen-

Scheme	Maps	Calls	Layers	Collision Resistance	Preimage Resistance
$\mathbf{LP}_{231}$	$2 \to 1$	3	3	$2^{0.5n}$	$2^{0.67n}$
$\mathbf{LP}_{241}$	$2 \to 1$	4	4	$2^{0.5n}$	$2^{0.75n}$
$\mathbf{LP}^{\mathbf{SS}}$	$2 \to 1$	3	2	$2^{0.50n}$	$2^{0.50n}$
SSt	$2 \to 1$	3	2	$2^{n/2}$	$2^{2n/3}$(conjectured)
eSSt	$2 \to 1$	4	3	$2^{n/2}$	2^n
$\mathbf{eSt}_{2n/3}$	$2 \to 1$	2	2	$2^{n/3}$	
GBe	$2 \to 1$	$3t$	2	$2^{n/2}$	$2^{\frac{t+1}{t+2}n}$(non-adaptive, adaptive conjectured)
$\mathbf{LP}_{352}$	$3 \to 2$	5	5	$2^{0.55n}$	$2^{0.80n}$
$\mathbf{LP}_{362}$	$3 \to 2$	6	6	$2^{0.63n}$	$2^{0.80n}$
StDL	$3 \to 2$	1	1	2^n(non-adaptive)	

Table 2.3: Comparison of Constructions. Maps shows the input/output size in multiples of n. CR and PR represents collsion resistance and preimage resistance respectively.

eral. Although an ideal cipher is structurally different from a random function, a fixed-key ideal cipher is merely a random permutation, so they are somewhat comparable in terms of efficiency. If the cost of calling a random permutation is the same as the cost of calling a random function, then **SSt** and **eSSt** seem to be better choices than $\mathbf{LP}_{231}$ and $\mathbf{LP}_{231}$ due to fewer layers.

eSSt is the only construction in the table which has both optimal collision resistance and preimage resistance, with a total of three layers making four calls. However, if the conjecture of **GBe** holds, then it also has nearly optimal preimage resistance while having only two layers. If circuit size is not a concern then **GBe** has an advantage over **eSSt** in performance due to parallel computing.

General Bounds

There is a powerful notion called *yield*, the number of evaluations to the compression function an adversary can make based on his/her query results from the underlying primitives. General bounds on collision resistance and preimage resistance of a compression function can be made based on this notion.

In [16], Rogaway and Steinberger introduced a condition called collision-uniformity. Given a hash function compression function $H : \{0,1\}^m \to \{0,1\}^n$ making k calls, define λ_H to be the smallest number such that there exists an adversary $\mathcal{A}$, who makes q queries with a yield of $\lambda_H 2^{n/2}$, such that the probability of $\mathcal{A}$ finding a collision for H is at least $1/2$. H is considered as collision-uniform if λ_H is a small constant. With the notion of yield and collision-uniformity, they showed the following bound:

Theorem 1. *Given H collision-uniform, a collsion can be found with constant probability for approximately $2^{(1-(m/n-0.5)/k)n}$ queries.*

Analogously, define δ_H to be the smallest number such that there exists an adversary $\mathcal{A}$, who makes q queries with a yield of $\delta_H 2^n$, such that the probability of $\mathcal{A}$ finding a preimage for H is at least $1/2$. H is considered as preimage-uniform if δ_H is a small constant. They showed another bound making use of yield and preimage-uniformity:

Theorem 2. *Given H preimage-uniform, a preimage can be found with constant probability for approximately $2^{(1-(m/n-1)/k)n}$ queries.*

However, not all compression functions have to behave like random functions. Stam showed that if a compression is not collision-uniform, then the bound does not hold [18]. The example he gave is exactly the construction $\mathbf{eSt}_{2n/3}$ (See Figure 2.9), which has a collision resistance of $2^{n/3}$ instead of $2^{(1-(2-0.5)/2)n} = 2^{n/4}$ queries. In general, the yield only shows the relationship between the number of H evaluations an adverary can at most make and the number of queries to the primitives.

Interestingly, Stam also proposed a bound about the yield [18].

Theorem 3. *If $H : \{0,1\}^{m+s} \to \{0,1\}^s$ is a compression function making one call to each of its r primitives $f_i : \{0,1\}^{n+c} \to \{0,1\}^n$, then there exists an adversary who can achieve a yield of at least $2^{m+s}(q/2^{n+c})^r$.*

He also gave a relation between the yield and collision resistance. In the same paper there is the following conjecture:

Conjecture 1. *If $H : \{0,1\}^{m+s} \to \{0,1\}^s$ is a compression function making r calls to $f : \{0,1\}^{n+c} \to \{0,1\}^n$, a collision can be found for $q \leq 2^{(nr+cr-m)/(r+1)}$.*

He also showed how the yield can obtain bounds on indifferentiability.

Theorem 4. *If $H : \{0,1\}^{m+s} \to \{0,1\}^s$ is a compression function making r calls to $f : \{0,1\}^{n+c} \to \{0,1\}^n$, then H is differentiable from a random function when $q > 2^{n+c}(\frac{n+r}{c} 2^{n+c-m-s})^{1/(r-1)}$.*

2.3.2 Existing Domain Extenders

For domain extenders, the *Strengthened Merkle-Damgård Construction* is known to preserve collision resistance. There are several ways to strengthen the plain construction (See Figure 2.2), and one fix, shown by Damgård himself [10], is presented in Figure 2.11:

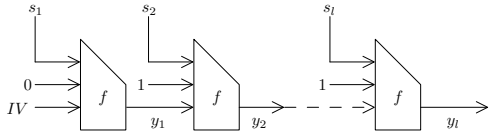

Figure 2.11: Strengthened Merkle-Damgård construction

Algorithm SMD$(s_1 \| \cdots \| s_l)$
$s_1 \leftarrow s_1 \| 0$
for $i = 2$ to l **do**
  $s_i \leftarrow s_i \| 1$
end for
$y \leftarrow$ **MD**$(s_1 \| \cdots \| s_l)$
return y

Besides preserving collision resistance, it also preserves everywhere-preimage resistance [2], but not for all the remaining six properties proposed by Rogaway and Shrimpton [14].

Because the goal is to design a hash function with as many properties as possible, there are domain extender designs aiming to preserve more properties from component functions. Andreeva et al. presented their *Random-Oracle-XOR (ROX) Construction* [2], preserving all seven hash function properties defined by Rogaway and Shrimpton in [14]. It uses two random oracles. One for the masks and the other for padding. See Figure 2.12 for details. Here

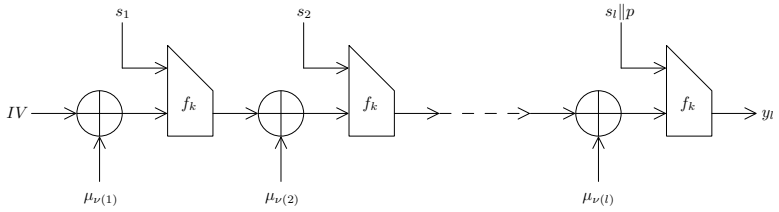

Figure 2.12: Random-Oracle-XOR construction

$\mu_{\nu(i)}$ is the mask and p is the padding. Let $O_1 : \{0,1\}^* \rightarrow \{0,1\}^n$ and $O_2 : \{0,1\}^* \rightarrow \{0,1\}^{2n}$ be random oracles. For key k and d being the first d bits of $s = s_1 \| \cdots \| s_l$, $\mu_{\nu(i)} = O_1(k, d, \nu(i)_{\text{bin}})$, where $\nu(i)$ is the largest integer j such that $2^j | i$. p is the padding output by the padding function rox-pad:

$$p = O_2(d, |s|_{\text{bin}}, 1_{\text{bin}}) \| O_2(d, |s|_{\text{bin}}, 2_{\text{bin}}) \| \cdots$$

with size at least $2n$, so it is possible to generate an extra block consisting of padding bits only. Here is the algorithm for the construction:

Algorithm ROX(k, s)
$s_1 \| \cdots \| s_l \leftarrow s \| \text{rox-pad}(s)$
$y_0 \leftarrow IV$
for $i = 0$ to $\lfloor \log_2(l) \rfloor$ **do**
  $\mu_i \leftarrow O_1(k, d, i_{\text{bin}})$
end for
for $i = 1$ to l **do**
  $g_i \leftarrow y_{i-1} \oplus \mu_{\nu(i)}$
  $y_i \leftarrow f_k(s_i \| g_i)$
end for
return y_l

In practice, compression functions are usually keyless, or contains a fixed built-in key. For these compression functions, Andreeva et al. proposed four keyless domain extenders which preserve collision, preimage and second preimage resistance [3]. Two of them are actually variants of ROX, and the other two are tree-based constructions.

Chapter 3

The Generalized Benes Construction

In this chapter, we first list results of the original Benes construction. We then introduce a class of compression functions, which we call *the Generalized Benes construction*, as well as giving some basic remarks regarding the construction.

3.1 The Benes Construction

The Benes construction, also called the double butterfly transformation, originates from the work of Aiello and Venkatesan [1]. It is a double-length scheme which yields a $2n$-bit to $2n$-bit function from n-bit to n-bit (private) random functions. In Figure 3.1, $F_1, \ldots, F_4$, $G_1, \ldots, G_4$ are n-bit to n-bit random functions. For input string $s_1 \| s_2$ where $s_1, s_2 \in \{0,1\}^n$,

1. The values $W_i = F_{2i-1}(s_1) \oplus F_{2i}(s_2)$ are computed for $i = 1, 2$.
2. The construction computes $Y_i = G_{2i-1}(W_1) \oplus G_{2i}(W_2)$ for $i = 1, 2$.
3. The output is $Y_1 \| Y_2$.

Given an information-theoretic adversary who can only query the whole construction as a black-box, the Benes construction is indistinguishable from a $2n$-bit to $2n$-bit random function for distinguishers making up to $\Omega(2^n)$ queries. Moreover, they showed that the construction is minimal by showing that deleting any edge in the diagram makes the resulting design vulnerable to birthday attacks using $O(2^{n/2})$ queries. However, there is a mistake in the proof of indistinguishability, and a complete correct proof is presented by Patarin [13].

The butterfly transformation itself also has interesting properties. Also mentioned by Aiello and Venkatesan [1], the butterfly transformation is similar to a Feistel transformation (See Figure 2.1). Both can connect itself in composition with seemingly increasing security. The Benes construction can hence be

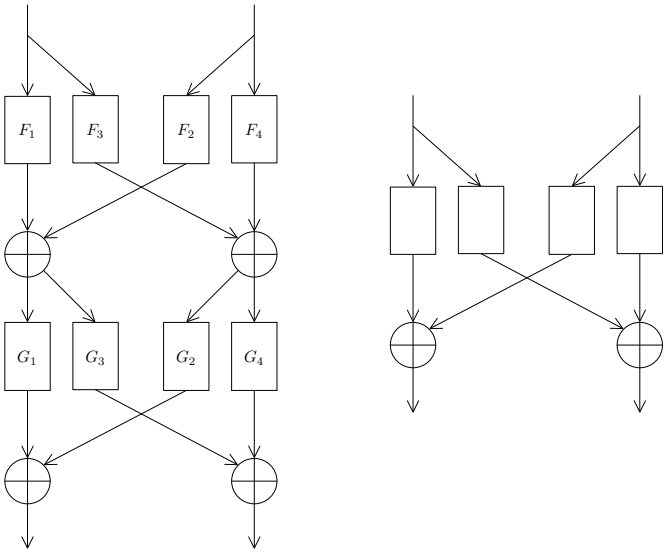

Figure 3.1: The Benes construction (left) and the butterfly transformation (right)

imagined as a two-round butterfly transformation. A difference between the two transformations is that the Feistel transformation is a permutation while the butterfly transformation is not. Aiello and Venkatesan [1] made a comparison between the Benes construction and a 4-round Feistel network. Although a 4-round Feistel transformation is still vulnerable to birthday attacks, thus only indistinguishable from a random function up to $O(2^{n/2})$ queries, it cannot be compared to the Benes construction directly since the number of random function calls and the number of layers are different. On one hand a butterfly transformation makes two calls to its underlying random functions while a Feistel transformation makes only one. On the other hand a round of Feistel transformation seems to provide less security than a round of butterfly transformation. Hence the butterfly transformation can serve as an alternative to the Feistel transformation.

When the Benes construction is put into the public random functions setting, where any adversary is allowed to query the underlying random functions, the security statement by Aiello and Venkatesan does not hold anymore. Maurer and Tessaro presented a distinguisher which can differentiate the Benes construction from a truly random function with constant probability while making only $O(2^{n/2})$ queries [12].

However, the analysis is far from done. Even though the Benes construction is less secure when the underlying random functions are public, if its variant has an output size of length n, such variant can still have optimal collision resistance and other useful properties. Therefore the Benes construction contains many possibilites and potentials, leading to our generalized design.

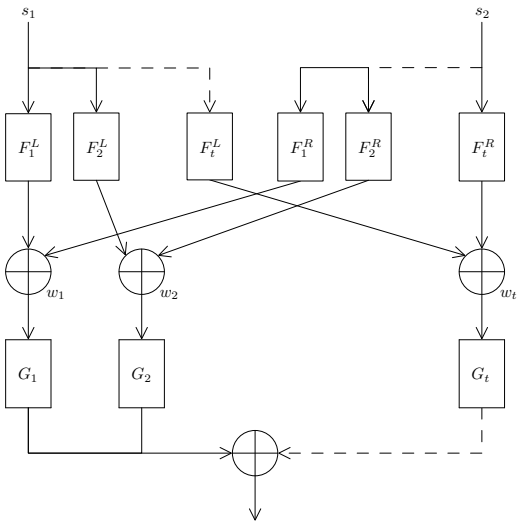

Figure 3.2: The Generalized Benes construction

3.2 The Generalized Benes Construction

Figure 3.2 shows our main construction, where $F_1^L, \ldots, F_t^L, F_1^R, \ldots, F_t^R, G_1, \ldots, G_t$ are independent n-to-n bit public random functions. Let $s_1 \| s_2$ be the input, where $s_1,\ s_2 \in \{0,1\}^n$. The output is computed in two stages:

1. For $i = 1, \ldots, t$, $W_i(s_1, s_2) = F_i^L(s_1) \oplus F_i^R(s_2)$ is computed.
2. $H_t(s_1 \| s_2) = \bigoplus_{i=1}^{t} G_i(W_i(s_1, s_2))$ is the output.

One can also express the construction in a compact form

$$H_t(s_1 \| s_2) = \bigoplus_{i=1}^{t} G_i(F_i^L(s_1) \oplus F_i^R(s_2))$$

Define $W(s_1, s_2) = W_1(s_1, s_2) \| \cdots \| W_t(s_1, s_2)$, the concatenation of values obtained after the first processing stage. Furthermore, define system G such that

$$G(W(s_1, s_2)) = \bigoplus_{i=1}^{t} G_i(W_i(s_1, s_2)) = H_t(s_1 \| s_2)$$

When put in words, G is the second processing stage of H_t, taking $W(s_1, s_2)$ and gives the final output $H_t(s_1 \| s_2)$. There are two remarks with respect to this construction.

- For $t = 2$, $W(s_1, s_2)$ is exactly the output of a butterfly transformation. H_2 is a slight modification of the Benes construction, with two random

functions removed and output merged by an exclusive-or operation, forming a compression function. This class of compression functions H_t is a generalization of the design of H_2.

- The properties of H_1 are very different from other compression functions in this class. It is very similar to the design by Shrimpton and Stam [17] but has weaker properties. It is therefore a degenerate case and will not be discussed in this thesis.

- It might seem unnecessary to make so many random function calls to achieve optimal collision resistance. The construction **SSt** by Shrimpton and Stam (See Figure 2.8) already suffices [17]. The reason for H_t making more calls is to provide better preimage resistance. One might then argue that the construction **eSSt** discussed in 2.3.1 is both optimally collision and preimage resistant while making only four calls. Note however, that **eSSt** is a three-layered design. Since H_t is a two-layered design, when both functions are implemented on hardware, H_t will run faster because its pipeline is shorter than that of **eSSt**.

Chapter 4

Collision Resistance of Generalizd Benes Construction

The core of this chapter is the proof of an upper bound on the collision-finding advantage of the Generalized Benes construction, as well as the interpretation part showing the maximum number of queries which the construction is secure up to. The proof resembles to the one given by Shrimpton and Stam for the collision resistance of **SSt** [17]. The proof structure is similar, but since the Generalized Benes Construction contains more random functions and is more difficult, the proof details and tricks used are different, so this proof is a non-trivial extension of their proof.

Starting from this chapter any adversary is considered to be information-theoretic, and adaptive unless explicitly specified. Under this assumption any attack is parameterized by $q(n)$, the number of queries to *each* underlying primitive function.

The random experiment is as follows: $F_1^L, \ldots, F_t^L$, $F_1^R, \ldots, F_t^R$, $G_1, \ldots, G_t$ are chosen from $\mathcal{F}_{n,n}$ uniformly at random. $\mathcal{A}$ can query up to q times to each random function in any order. Finally, $\mathcal{A}$ has to output two strings $s_1 \| s_2$ and $s_1' \| s_2'$, and he/she wins the game if the two strings are distinct and $H_t(s_1 \| s_2) = H_t(s_1' \| s_2')$.

Queries which $\mathcal{A}$ makes can be divided into three types:

1. (F_i^L, s) denotes a query to F_i^L with input s for $1 \leq i \leq t$. Such query will be answered by $F_i^L(s)$.

2. (F_i^R, s) denotes a query to F_i^R with input s for $1 \leq i \leq t$. Such query will be answered by $F_i^R(s)$.

3. (G_i, w) denotes a query to G_i with input w for $1 \leq i \leq t$. Such query will be answered by $G_i(w)$.

Let $Q_{\mathcal{A}}$ be the set of queries $\mathcal{A}$ has made until the moment when he/she outputs the two strings. We say $H_t(s_1, s_2)$ is computable from $Q_{\mathcal{A}}$ if (F_i^L, s_1), (F_i^R, s_2), $(G_j, F_i^L(s_1) \oplus F_i^R(s_2)) \in Q_{\mathcal{A}}$ for all $1 \leq i, j \leq t$. We shall refine the definition of collision-finding advantage in Section 2.2.1 to be more specific:

Definition 2. *The collision-finding advantage of $\mathcal{A}$ with respect to the Generalized Benes construction H_t is defined as*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{Coll}}_{H_t(n)}(\mathcal{A}) = \Pr\Big[&F_1^L, \ldots, F_t^L, F_1^R, \ldots, F_t^R, G_1, \ldots, G_t \xleftarrow{\$} \mathcal{F}_{n,n};\\ &s_1\|s_2, s_1'\|s_2' \leftarrow \mathcal{A}^{F_1^L,\ldots,F_t^L,F_1^R,\ldots,F_t^R,G_1,\ldots,G_t} :\\ &s_1\|s_2 \neq s_1'\|s_2', H_t(s_1\|s_2) = H_t(s_1'\|s_2'),\ \textit{and both}\\ &H_t(s_1, s_2)\ \textit{and}\ H_t(s_1', s_2')\ \textit{are computable from}\ Q_{\mathcal{A}}\Big]\end{aligned}$$

Note that such definition, for convenience, prevents $\mathcal{A}$ from guessing. We shall now state the upper bound of the collision-finding advantage of any adversary making q queries here:

Theorem 5. *Let $\mathcal{A}$ be an adversary making q queries to every underlying random function of H_t, then for $t \geq 2$ and $k \geq 2$,*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{Coll}}_{H_t(n)}(\mathcal{A}) \leq (tq)^2\frac{(tq)^2-1}{2}2^{-tn} + tq(tq-1)2^{-n} + \frac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!}\\ + \frac{k^2(q-1)q}{2}2^{-n} + q\binom{k}{2}2^{-n}\end{aligned}$$

The remaining sections are dedicated to the proof of the theorem, with the outline of the proof described in Section 4.1.

4.1 Proof Preparation

Before we go into the details of the proof, there are several definitions and key observations which can greatly simplify the random experiment, and are essential for the proof.

Observation 1 Let $s_1\|s_2$ and $s_1'\|s_2'$ be distinct strings. If $W(s_1, s_2) = W(s_1', s_2')$, then $H_t(s_1\|s_2) = H_t(s_1'\|s_2')$ for sure, since

$$H_t(s_1\|s_2) = G(W(s_1, s_2)) = G(W(s_1', s_2')) = H_t(s_1'\|s_2')$$

Note that the converse is not true though, so collisions can be divided into two types, defined as follows:

Definition 3. *A pair of inputs $s_1\|s_2$ and $s_1'\|s_2'$ cause an* internal collision *if $W(s_1, s_2) = W(s_1', s_2')$. $s_1\|s_2$ and $s_1'\|s_2'$ cause a* final collision *if $H_t(s_1\|s_2) = H_t(s_1'\|s_2')$ but $W(s_1, s_2) \neq W(s_1', s_2')$.*

Hence any collision for H_t is either internal or final.

Observation 2 Since the functions $F_1^L, \ldots, F_t^L, F_1^R, \ldots, F_t^R$ reply queries with uniform random strings as long as $\mathcal{A}$ does not repeat a query, querying adaptively does not help. We can exploit this property to simplify the random experiment.

Assume when $\mathcal{A}$ queries F_i^L with the input s, the results $F_j^L(s)$ are also given to $\mathcal{A}$ for all $j \neq i$ for free. Similarly, when $\mathcal{A}$ queries F_i^R with the input s, the results $F_j^R(s)$ are given to $\mathcal{A}$ for all $j \neq i$. Since $\mathcal{A}$ can choose to ignore the extra information, his/her collision-finding advantage can only increase. Under such assumption, $\mathcal{A}$ can have at most tq query results from F_i^L for $1 \leq i \leq t$. The same holds for F_i^R as well.

Given $\mathcal{A}$ can only get at most tq query results from F_i^L and F_i^R for $1 \leq i \leq t$, F_i^L and F_i^R can be replaced by lists of tq random n-bit strings without changing the distribution of query results $\mathcal{A}$ gets. The values on the list are not associated to any particular input. Every time $\mathcal{A}$ queries F_i^L or F_i^R with a new input, he/she can just fetch a new value from the list. We can even assume $\mathcal{A}$ receives all $2t$ lists beforehand.

Based on this observation, the random experiment can be simplified into the following: $G_1, \ldots, G_t$ are chosen uniformly at random from $\mathcal{F}_{n,n}$. $\mathcal{A}$ receives random tn-bit strings $X_1, \ldots, X_{tq}, Y_1, \ldots, Y_{tq}$ at the beginning, before any query is made. He/She can then make q queries to each of the random functions $G_1, \ldots, G_t$ in any order. Finally he/she outputs two distinct pairs $(i, j), (i', j')$ where $1 \leq i, i', j, j' \leq tq$ and $G(X_i \oplus Y_j)$, $G(X_{i'} \oplus Y_{j'})$ are computable from $Q_{\mathcal{A}}$. $\mathcal{A}$ wins the game if $X_i \oplus Y_j = X_{i'} \oplus Y_{j'}$, or $G(X_i \oplus Y_j) = G(X_{i'} \oplus Y_{j'})$. Call this game *Game1* and let $\mathbf{Adv}_{H_t(n)}^{Game1}(\mathcal{A})$ be the probability that $\mathcal{A}$ wins in *Game1*.

Lemma 2. *Given any adversary $\mathcal{A}$, there exists an adversary $\mathcal{A}'$ such that*

$$\mathbf{Adv}_{H_t(n)}^{\mathrm{Coll}}(\mathcal{A}) \leq \mathbf{Adv}_{H_t(n)}^{Game1}(\mathcal{A}')$$

Proof. The proof is to build $\mathcal{A}'$ using $\mathcal{A}$ as a component. Split X_i and Y_j into t n-bit string blocks. Let $X_i^{(k)}$ and $Y_j^{(k)}$ be the kth block of X_i and Y_j respectively. Given $\mathcal{A}$, if $\mathcal{A}$ makes the query (F_k^L, s), $\mathcal{A}'$ does the following:

1. $\mathcal{A}'$ searches for an i such that $X_i^{(k)}$ is associated to s.
2. If such i is found, then (F_k^L, s) has been queried before. Return $X_i^{(k)}$ to $\mathcal{A}$.
3. Otherwise, (F_k^L, s) is a new query. $\mathcal{A}'$ finds a minimum i such that $X_i^{(k)}$ is not associated to any string, then $\mathcal{A}'$ associate s to $X_i^{(k)}$ and return $X_i^{(k)}$ to $\mathcal{A}$.

Note that all blocks have no association with any string before $\mathcal{A}$ starts querying. If $\mathcal{A}$ makes the query (F_k^R, s), $\mathcal{A}'$ reacts similarly:

1. $\mathcal{A}'$ searches for a j such that $Y_j^{(k)}$ is associated to s.
2. If such j is found, then return $Y_j^{(k)}$ to $\mathcal{A}$.

3. Otherwise, $\mathcal{A}'$ finds a minimum j such that $Y_j^{(k)}$ is not associated to any string, then $\mathcal{A}'$ associate s to $Y_j^{(k)}$ and return $Y_j^{(k)}$ to $\mathcal{A}$.

If $\mathcal{A}$ makes the query (G_i, s), $\mathcal{A}'$ forwards the query by querying G_i with the inputs s. $\mathcal{A}'$ then returns the query result to $\mathcal{A}$.

Finally suppose $\mathcal{A}$ outputs two strings $s_1 \| s_2$ and $s_1' \| s_2'$. Since $H_t(s_1, s_2)$ and $H_t(s_1', s_2')$ are computable from $Q_{\mathcal{A}}$ by definition, all necessary queries are made and processed by $\mathcal{A}'$. Therefore $\mathcal{A}'$ can search for an i and a j such that $X_i^{(1)}$ is associated to s_1 and $Y_j^{(1)}$ is associated to s_2. Similarly, $\mathcal{A}'$ can search for an i' and a j' such that $X_{i'}^{(1)}$ and $Y_{j'}^{(1)}$ are associated to s_1' and s_2' respectively. $\mathcal{A}'$ can then output the two pairs (i, j) and (i', j').

By inspection, it should be clear that $\mathcal{A}'$ wins in *Game1* with at least the same probability as $\mathcal{A}$ winning in the original random experiment. □

By the lemma above, we can focus on $\mathbf{Adv}_{H_t(n)}^{Game1}(\mathcal{A})$ where $\mathcal{A}$ still has q queries for every random function $G_1, \ldots, G_t$. We shall investigate the random functions $G_1, \ldots, G_t$ to further simplify the random experiment.

Observation 3 If no two pairs (i, j) and (i', j') exist such that $X_i \oplus Y_j = X_{i'} \oplus Y_{j'}$, then $\mathcal{A}$ will need to query $G_1, \ldots, G_t$ in order to win. Since $\mathcal{A}$ can query the functions adaptively, which makes analysis difficult, we can further assume that results $G_k(X_i^{(k)} \oplus Y_j^{(k)})$ are given to $\mathcal{A}$, and $(G_k, X_i^{(k)} \oplus Y_j^{(k)}) \in Q_{\mathcal{A}}$ for all $1 \leq i, j \leq tq$ and $2 \leq k \leq t$. Hence there is no need for $\mathcal{A}$ to make any query to $G_2, \ldots, G_t$ and can focus on querying G_1. We shall prove that $\mathcal{A}$ still needs many queries to find a final collision.

The resulting random experiment is similar to *Game1*, but with $G_k(X_i^{(k)} \oplus Y_j^{(k)})$ given to $\mathcal{A}$, and $(G_k, X_i^{(k)} \oplus Y_j^{(k)}) \in Q_{\mathcal{A}}$ for all $1 \leq i, j \leq tq$ and $2 \leq k \leq t$. Since this is equivalent to $\mathcal{A}$ in *Game1* but with extra information, his/her chances to win can only increase. Name this random experiment *Game2*.

Observation 4 There is a very useful lemma stated by Shrimpton and Stam [17], which states:

Lemma 3. *Let A and B be distributions induced by sampling from $\{0,1\}^n$ without replacement. Let $\mathbf{a}$ and $\mathbf{b}$ be vectors of size q with elements drawn according to A and B respectively. Let $\mathrm{kcoll}_{\mathbf{a} \oplus \mathbf{b}}$ be the event that there exists a k-way collision in the tensor product vector $(\mathbf{a} \otimes \mathbf{b})$ under exclusive-or, then*

$$\Pr\left[\mathrm{kcoll}_{\mathbf{a} \oplus \mathbf{b}}\right] \leq \frac{(q!)^2 2^n (2^n - k)!}{((q-k)!)^2 k! (2^n)!}$$

The only obstacle which prevents us from using the lemma is the fact that query answers from a random function are equivalent to sampling *with* replacement. However, if a collision cannot be identified from query replies, a random function is indistinguishable from a random permutation. Therefore, given a random function answers just like a random permutation, i.e. no two distinct

query inputs are replied with the same answer, it is indistinguishable from a random permutation and we can apply the lemma. In Section 4.3 this lemma will prove to be very important.

We are ready to start proving theorem 5. Consider $\mathbf{Adv}^{Game2}_{H_t(n)}(\mathcal{A})$. Let icoll be the event that there exists distinct pairs $(i,j),(i',j')$ such that $X_i \oplus Y_j = X_{i'} \oplus Y_{j'}$. Let kcoll_{W_1} be the event that there is a k-way collision in the list $X_i^{(1)} \otimes Y_j^{(1)}$.

The collision-finding advantage of $\mathcal{A}$ can be upper bounded based on *Game2*.

$$\begin{aligned}
\mathbf{Adv}^{\text{Coll}}_{H_t(n)}(\mathcal{A}) &\leq \mathbf{Adv}^{Game2}_{H_t(n)}(\mathcal{A}) \\
&\leq \Pr[\text{icoll}] \cdot 1 + (1 - \text{icoll})\mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \,\middle|\, \overline{\text{icoll}}\right) \\
&\leq \Pr[\text{icoll}] + \mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \,\middle|\, \overline{\text{icoll}}\right) \\
&\leq \Pr[\text{icoll}] + \Pr[\text{kcoll}_{W_1}] + \mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \,\middle|\, \overline{\text{icoll}}, \overline{\text{kcoll}_{W_1}}\right) \quad (4.1)
\end{aligned}$$

The proof of Theorem 5 will be divided into three steps:

1. Upper bounding Pr[icoll] (Section 4.2).
2. Upper bounding $\Pr[\text{kcoll}_{W_1}]$ (Section 4.3).
3. Upper bounding $\mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \,\middle|\, \overline{\text{icoll}}, \overline{\text{kcoll}_{W_1}}\right)$ (Section 4.4).

Every step will be done in a separate subsection.

4.2 Bounding Pr[icoll]

Bounding Pr[icoll] is straight forward.

Lemma 4.

$$\Pr[\text{icoll}] \leq (tq)^2 \frac{(tq)^2 - 1}{2} 2^{-tn}$$

For any distinct fixed pairs $(i,j),(i',j')$, $X_i \oplus Y_j = X_{i'} \oplus Y_{j'}$ with probability 2^{-tn}. Since there are $((tq)^4 - (tq)^2)/2$ ways to choose $(i,j),(i',j')$, by the union bound we have

$$\Pr[\text{icoll}] \leq (tq)^2 \frac{(tq)^2 - 1}{2} 2^{-tn}$$

Substituting this results into inequality (4.1) eliminates one unknown term:

$$\begin{aligned}
\mathbf{Adv}^{\text{Coll}}_{H_t(n)}(\mathcal{A}) \leq (tq)^2 \frac{(tq)^2 - 1}{2} 2^{-tn} &+ \Pr[\text{kcoll}_{W_1}] \\
&+ \mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \,\middle|\, \overline{\text{icoll}}, \overline{\text{kcoll}_{W_1}}\right) \quad (4.2)
\end{aligned}$$

4.3 Bounding $\Pr[\text{kcoll}_{W_1}]$

This section is dedicated to bounding $\Pr[\text{kcoll}_{W_1}]$.

Lemma 5.

$$\Pr[\text{kcoll}_{W_1}] \leq tq(tq-1)2^{-n} + \frac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!}$$

Proof. Let $\text{coll}_{X_i^{(1)}}$ be the event that $X_i^{(1)} = X_{i'}^{(1)}$ for some distinct $1 \leq i, i' \leq tq$. Let $\text{coll}_{Y_j^{(1)}}$ be the event that $Y_j^{(1)} = Y_{j'}^{(1)}$ for some distinct $1 \leq j, j' \leq tq$. Given both $\text{coll}_{X_i^{(1)}}$ and $\text{coll}_{Y_j^{(1)}}$ do not hold, $X_i^{(1)}$ and $Y_j^{(1)}$ are equivalent to sampling from $\{0,1\}^n$ without replacement, so Lemma 3 can be applied. Assume $k \geq 2$.

$$\begin{aligned}
\Pr[\text{kcoll}_{W_1}] &\leq \Pr\left[\text{coll}_{X_i^{(1)}} \cup \text{coll}_{Y_j^{(1)}}\right] + \left(1 - \Pr\left[\text{coll}_{X_i^{(1)}} \cup \text{coll}_{Y_j^{(1)}}\right]\right) \\
&\quad \Pr\left[\text{kcoll}_{W_1} \left| \overline{\text{coll}_{X_i^{(1)}}}, \overline{\text{coll}_{Y_j^{(1)}}} \right.\right] \\
&\leq \Pr\left[\text{coll}_{X_i^{(1)}}\right] + \Pr\left[\text{coll}_{Y_j^{(1)}}\right] + \Pr\left[\text{kcoll}_{W_1} \left| \overline{\text{coll}_{X_i^{(1)}}}, \overline{\text{coll}_{Y_j^{(1)}}} \right.\right] \\
&\leq \binom{tq}{2} 2^{-n} + \binom{tq}{2} 2^{-n} + \frac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!} \\
&= tq(tq-1)2^{-n} + \frac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!}
\end{aligned}$$

□

Integrating this result into (4.1) gives

$$\begin{aligned}
\mathbf{Adv}^{\text{Coll}}_{H_t(n)}(\mathcal{A}) \leq (tq)^2 \frac{(tq)^2 - 1}{2} 2^{-tn} + tq(tq-1)2^{-n} + \frac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!} \\
+ \mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \left| \overline{\text{icoll}}, \overline{\text{kcoll}_{W_1}} \right.\right) \qquad (4.3)
\end{aligned}$$

4.4 Bounding Final-Collision-Finding Advantage

The current goal is to find $\mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \left| \overline{\text{icoll}}, \overline{\text{kcoll}_{W_1}} \right.\right)$. This is a probability conditioned on $\overline{\text{icoll}}$ and $\overline{\text{kcoll}_{W_1}}$, so we will assume that

- $\overline{\text{icoll}}$ holds, i.e. no distinct pairs $(i,j),(i',j')$ exists such that $X_i \oplus Y_j = X_{i'} \oplus Y_{j'}$. Let set $S_W = \{X_i \oplus Y_j | 1 \leq i, j \leq tq\}$.
- $\overline{\text{kcoll}_{W_1}}$ holds for $k \geq 2$, i.e. there are no k-way collisions in the list $X_i^{(1)} \otimes Y_j^{(1)}$, so for any $w \in \{0,1\}^n$,

$$|\{w_1 \| \cdots \| w_t \in S_W | w_1 = w\}| \leq k$$

The notion of yield is introduced in Section 2.2. Here we shall define it formally with respect to *Game2*:

Definition 4. *Let $S \subseteq \{0,1\}^{tn}$, then the yield of S is*

$$\text{yield}(S) = \max_{\substack{S^* \subseteq \{0,1\}^n \\ |S^*|=q}} |\{w_1 \| \cdots \| w_t \in S | w_1 \in S^*\}|$$

All $\mathcal{A}$ needs to do is to query G_1. Although he/she can still make queries to $G_2, \ldots, G_t$, with the extra information he/she has in *Game2* this is pointless. $G(X_i \oplus Y_j)$ is computable from $Q_{\mathcal{A}}$ if and only if $(G_1, X_i^{(1)} \oplus Y_j^{(1)}) \in Q_{\mathcal{A}}$. Hence every time $\mathcal{A}$ makes a query to G_1, he/she will be able to evaluate G on some more elements in S_W. Let S_i be the set of strings in S_W which $\mathcal{A}$ can evaluate G on after the ith query to G_1 (before the $(i+1)$th query), i.e. with respect to queries $\mathcal{A}$ have sent right after the ith query to G_1,

$$S_i = \{X_{i'} \oplus Y_{j'} | (G_1, X_{i'}^{(1)} \oplus Y_{j'}^{(1)}) \text{ is queried } \}$$

Let $e_i = |S_i \backslash S_{i-1}|$, then right after $\mathcal{A}$ has made the ith query to G_1, he/she will be able to evaluate G on e_i more elements in S_W.

If $\mathcal{A}$ can find a collision, then there exists a unique $1 \leq i \leq q$ such that S_i contains a collision but not S_{i-1}. Given there is no colliding pair in S_{i-1}, S_i contains a collision if $S_i \backslash S_{i-1}$ contains a colliding pair, or $S_i \backslash S_{i-1}$ has a string colliding with some other string in S_{i-1}.

Since $e_i \leq k$ by assumption $\overline{\text{kcoll}_{W_1}}$, $S_i \backslash S_{i-1}$ contains a colliding pair with probability at most $\binom{k}{2} 2^{-n}$ for any $1 \leq i \leq q$. $S_i \backslash S_{i-1}$ has size e_i and $|S_{i-1}| = \sum_{j=1}^{i-1} e_j$, therefore $S_i \backslash S_{i-1}$ has a string colliding with some other string in S_{i-1} with probability at most $2^{-n} e_i \sum_{j=1}^{i-1} e_j$. Combining the two probabilities and applying the union bound over all $1 \leq i \leq q$ gives:

$$\begin{aligned}
\mathbf{Adv}_{H_t(n)}^{Game2}\left(\mathcal{A} \middle| \overline{\text{icoll}}, \overline{\text{kcoll}_{W_1}}\right) &\leq 2^{-n} \sum_{i=1}^{q} e_i \sum_{j=1}^{i-1} e_j + \sum_{i=1}^{q} \binom{k}{2} 2^{-n} \\
&= 2^{-n} \sum_{i=1}^{q} e_i \sum_{j=1}^{i-1} e_j + q \binom{k}{2} 2^{-n}
\end{aligned}$$

The exact values $e_1, \ldots, e_q$ depend on the set of queries $\mathcal{A}$ made to G_1, so the bound is not very useful. To derive a better bound, we need the following two lemmas:

Lemma 6. *For $e_1, \ldots, e_q$ as above,*

$$\sum_{i=1}^{q} e_i \leq \text{yield}(S_W)$$

Proof. Consider yield(S_W). By definition there exists a set $S^* \subseteq \{0,1\}^n$, where $|S^*| = q$, such that $|\{w_1 \| \cdots \| w_t \in S | w_1 \in S^*\}|$ is maximized. If $\mathcal{A}$ uses S^* as the set of queries to G_1, then

$$\sum_{i=1}^{q} e_i = \text{yield}(S_W)$$

On the other hand, if $\mathcal{A}$ can query G_1 such that $\sum_{i=1}^{q} e_i > \text{yield}(S_W)$, then by setting S^* to be the set of queries $\mathcal{A}$ sent to G_1, $|\{w_1 \| \cdots \| w_t \in S | w_1 \in S^*\}| > \text{yield}(S_W)$, contradicting its definition. □

Lemma 7. *Suppose $e_1, \ldots, e_q$ are nonnegative real numbers such that $\sum_{i=1}^{q} e_i = y$, then $\sum_{i=1}^{q} e_i \sum_{j=1}^{i-1} e_j$ reaches its maximum if $e_1 = \cdots = e_q$.*

Proof. This can be proved by using Lagrange multipliers. Define

$$f(e_1, \ldots, e_q) = \sum_{i=1}^{q} e_i \sum_{j=1}^{i-1} e_j$$
$$g(e_1, \ldots, e_q) = \sum_{i=1}^{q} e_i$$

$$\begin{aligned} \nabla f &= \left(\sum_{i=1}^{q} e_i - e_1, \sum_{i=1}^{q} e_i - e_2, \ldots, \sum_{i=1}^{q} e_i - e_q \right) \\ &= (y - e_1, y - e_2, \ldots, y - e_q) \\ \nabla g &= (\underbrace{1, \ldots, 1}_{q}) \end{aligned}$$

For λ being the Lagrange multiplier, we have

$$\nabla f = \lambda \nabla g$$

Expanding this system of equations gives

$$y - e_i = \lambda$$

for all $i = 1, \ldots, q$, with a unique solution of $e_1 = \cdots = e_q = \frac{y}{q}$ and $\lambda = \frac{q-1}{q} y$. This solution gives either a maximum or minimum, but since $f(0, \ldots, 0, 1) = 0$ and $f\left(\frac{y}{q}, \ldots, \frac{y}{q}\right) > 0$. This solution maximizes f. □

Although $e_1, \ldots, e_q$ have to be integers, the result of Lemma 7 can definitely be used as an upper bound. Substituting $e_q = \frac{y}{q}$ back to the expression gives

$$\begin{aligned}
\sum_{i=1}^{q} e_i \sum_{j=1}^{i-1} e_j &\leq \sum_{i=1}^{q} \frac{y}{q}(i-1)\frac{y}{q} \\
&= \frac{y^2}{q^2} \sum_{i=1}^{q} (i-1) \\
&= \frac{y^2}{q^2} \sum_{i=1}^{q-1} i \\
&= \frac{y^2}{q^2} \frac{q(q-1)}{2} \\
&= \frac{(q-1)y^2}{2q}
\end{aligned}$$

By Lemma 6, the right hand side is maximized when $y = \mathrm{yield}(S_W)$. Hence we have a new bound for $\mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \left| \overline{\mathrm{icoll}}, \overline{\mathrm{kcoll}_{W_1}} \right.\right)$.

$$\begin{aligned}
\mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \left| \overline{\mathrm{icoll}}, \overline{\mathrm{kcoll}_{W_1}} \right.\right) &\leq 2^{-n} \sum_{i=1}^{q} e_i \sum_{j=1}^{i-1} e_j + q\binom{k}{2}2^{-n} \\
&\leq 2^{-n} \frac{(q-1)\,\mathrm{yield}(S_W)^2}{2q} + q\binom{k}{2}2^{-n}
\end{aligned}$$

By assumption, $|\{w_1 \| \cdots \| w_t \in S_W | w_1 = w\}| \leq k$ for any $w \in \{0,1\}^n$, so $\mathrm{yield}(S_W) \leq kq$. Finally we have the bound:

$$\mathbf{Adv}^{Game2}_{H_t(n)}\left(\mathcal{A} \left| \overline{\mathrm{icoll}}, \overline{\mathrm{kcoll}_{W_1}} \right.\right) \leq \frac{k^2(q-1)q}{2}2^{-n} + q\binom{k}{2}2^{-n}$$

Integrating this result into (4.1) completes the proof of Theorem 5.

4.5 Interpretation of Theorem 5

Corollary 1. *Suppose $t \geq 2$, then for any constant $\epsilon > 0$, $\mathbf{Adv}^{\mathrm{Coll}}_{H_t(n)}(\mathcal{A})$ is negligible for $q = 2^{\left(\frac{1}{2}-\epsilon\right)n}$.*

Proof. By Theorem 5 we have

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{Coll}}_{H_t(n)}(\mathcal{A}) &\le (tq)^2\frac{(tq)^2-1}{2}2^{-tn} + tq(tq-1)2^{-n} + \frac{((tq)!)^2 2^n(2^n-k)!}{((tq-k)!)^2 k!(2^n)!} \\
&\quad + \frac{k^2(q-1)q}{2}2^{-n} + q\binom{k}{2}2^{-n} \\
&\le (tq)^4 2^{-tn} + (tq)^2 2^{-n} + k^2q^2 2^{-n} + qk^2 2^{-n} \\
&\quad + \frac{((tq)!)^2 2^n(2^n-k)!}{((tq-k)!)^2 k!(2^n)!} \\
&\le t^4\left(\frac{q^2}{2^n}\right)^2 + t^2\frac{q^2}{2^n} + k^2\frac{q^2}{2^n} + k^2\frac{q}{2^n} \\
&\quad + \frac{((tq)!)^2 2^n(2^n-k)!}{((tq-k)!)^2 k!(2^n)!}
\end{aligned}$$

for any $k \ge 2$. Set $k = n$. For $q = 2^{\left(\frac{1}{2}-\epsilon\right)n}$,

$$\begin{aligned}
&t^4\left(\frac{q^2}{2^n}\right)^2 + t^2\frac{q^2}{2^n} + k^2\frac{q^2}{2^n} + k^2\frac{q}{2^n} \\
&\le t^4 2^{-4\epsilon n} + y^2 2^{-2\epsilon n} + n^2 2^{-2\epsilon n} + n^2 2^{\left(-\frac{1}{2}-\epsilon\right)n}
\end{aligned}$$

To show that $\dfrac{(tq!)^2 2^n(2^n-k)!}{((tq-k)!)^2 k!(2^n)!}$ is negligible, consider its natural logarithm and apply Stirling's Formula:

$$h! \approx \sqrt{2\pi h}\left(\frac{h}{e}\right)^h$$

$$\begin{aligned}
&(2tq+1)\ln(tq) - 2tq + n\ln 2 + \left(2^n - k + \frac{1}{2}\right)\ln\left(2^n-k\right) - (2^n-k) \\
&+ 2(tq-k) - (2(tq-k)+1)\ln\left(tq-k\right) - (2^n+\frac{1}{2})\ln 2^n + 2^n - k\ln k \\
&+ k - \frac{1}{2}(\ln k + \ln 2\pi) \\
&\le (2tq+1)\ln(tq) + n\ln 2 - nk\ln 2 - (2(tq-k)+1)\ln\left(tq-k\right) - k\ln k \\
&- \frac{1}{2}(\ln k + \ln 2\pi) \\
&\le 2k\ln(tq) + n\ln 2 - nk\ln 2 - k\ln k - \frac{1}{2}(\ln k + \ln 2\pi) \\
&= 2k\ln t + (nk - 2nk\epsilon)\ln 2 + n\ln 2 - nk\ln 2 - k\ln k - \frac{1}{2}(\ln k + \ln 2\pi) \\
&= 2n\ln t + n\ln 2 - 2n^2\epsilon\ln 2 - n\ln n - \frac{1}{2}(\ln n + \ln 2\pi)
\end{aligned}$$

Here $-n\ln n$ is the dominating term, so

$$\frac{(q!)^2 2^n(2^n-k)!}{((q-k)!)^2 k!(2^n)!} \le e^{-n\ln(n)}$$

which is negligible. □

Another interpretation of the asymtotic behavior of the Generalized Benes construction is shown by

Corollary 2. *Suppose $t \geq 2$, then for any constant $c > 1$,*

$$\lim_{n\to\infty} \mathbf{Adv}^{\mathrm{Coll}}_{H_t(n)}(\mathcal{A}) = 0$$

for $q = O(2^{n/2}/n^c)$.

Proof. By Theorem 5 we have

$$\mathbf{Adv}^{\mathrm{Coll}}_{H_t(n)}(\mathcal{A}) \leq t^4 \left(\frac{q^2}{2^n}\right)^2 + t^2\frac{q^2}{2^n} + k^2\frac{q^2}{2^n} + k^2\frac{q}{2^n} + \frac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!}$$

for any $k \geq 2$. Set $k = n$. Let $d > 0$ be a constant, then for $q = d2^{n/2}/n^c$,

$$\lim_{n\to\infty} \frac{q^2}{2^{-n}} = \lim_{n\to\infty} \frac{d^2}{n^{2c}} = 0$$
$$\lim_{n\to\infty} k^2\frac{q^2}{2^{-n}} = \lim_{n\to\infty} \frac{d^2}{n^{2c-2}} = 0$$
$$\lim_{n\to\infty} \left(t^4 \left(\frac{q^2}{2^n}\right)^2 + t^2\frac{q^2}{2^n} + k^2\frac{q^2}{2^n} + k^2\frac{q}{2^n}\right) = 0$$

To show that $\dfrac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!} \to 0$, consider its natural logarithm and apply Stirling's Formula:

$$\frac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!} \leq 2k\ln(tq) + n\ln 2 - nk\ln 2 - k\ln k - \frac{1}{2}(\ln k + \ln 2\pi)$$

Substituting $k = n$ and $q = d2^{n/2}/n^c$ into the inequality leads to:

$$\begin{aligned}
&2n\left(\ln t + \frac{n}{2}\ln 2 - c\ln n + \ln d\right) + n\ln 2 - n^2\ln 2 - n\ln n - \frac{1}{2}(\ln n + \ln 2\pi) \\
&\leq -2nc\ln n + n(2\ln t + \ln 2 + 2\ln d) - n\ln n - \frac{1}{2}(\ln n + \ln 2\pi) \\
&\leq -(2c+1)n\ln n + n(2\ln t + \ln 2 + 2\ln d) - \frac{1}{2}(\ln n + \ln 2\pi)
\end{aligned}$$

Here $-(2c+1)n\ln n$ is the dominating term, which tends to negative infinity, so

$$\lim_{n\to\infty} \frac{((tq)!)^2 2^n (2^n-k)!}{((tq-k)!)^2 k! (2^n)!} = 0$$

□

Chapter 5

Preimage Resistance of Generalized Benes Construction

In this chapter we conduct an analysis on the preimage resistance of the Generalized Benes construction. We reduce the problem of bounding the preimage-finding advantage of an adversary into an isolated mathematical problem. Remarks about the relationship of the reduced problem and the preimage resistance of Shrimpton and Stam's construction [17] are also given. We then present our main result: preimage resistance against non-adaptive adversaries and a preimage attack. Though not important in the analysis of collision resistance, by adjusting the parameter t one can raise the preimage resistance of the construction against non-adaptive adversaries arbitrarily close to 2^n queries. At last we have several suggestions of research approaches will may lead to bounds against adaptive adversaries.

Again we start by refining the preimage-finding advantage of $\mathcal{A}$, defined in Section 2.2.1, for our construction:

Definition 5. *The preimage-finding advantage of $\mathcal{A}$ with respect to the Generalized Benes construction H_t is defined as*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{Pre}}_{H(n)}(\mathcal{A}) = \Pr\Big[&F_1,\ldots,F_{2t},G_1,\ldots,G_t \xleftarrow{\$} \mathcal{F}_{n,n};\\ &y \leftarrow \mathcal{A}(); s_1\|s_2 \leftarrow \mathcal{A}^{F_1,\ldots,F_{2t},G_1,\ldots,G_t} :\\ &H_t(s_1\|s_2) = y \text{ and } H_t(s_1,s_2) \text{ is computable from } Q_{\mathcal{A}}\Big]\end{aligned}$$

Note that this definition corresponds to the everywhere-preimage-finding advantage in Section 2.2.1, but again, for convenience, we add an additional constraint forcing $\mathcal{A}$ to evaluate $H_t(s_1, s_2)$. Note that $\Pr[H_t(s_1\|s_2) = y] = 2^{-n}$ for any $y, s_1, s_2 \in \{0,1\}^n$.

Similar to the proof of collision resistance in Section 4.1, the random experiment of finding a preimage can be simplified. Define *Game3* to be the following random experiment: $G_1, \ldots, G_t$ are chosen uniformly at random from $\mathcal{F}_{n,n}$. $\mathcal{A}$ receives random tn-bit strings $X_1, \ldots, X_{tq}$, $Y_1, \ldots, Y_{tq}$ at the beginning, before any queries are made. He/She can then make q queries to each of the random functions $G_1, \ldots, G_t$ in any order. Finally he/she outputs a pair (i, j) where $1 \leq i, j \leq tq$ and $G(X_i \oplus Y_j)$ is computable from $Q_{\mathcal{A}}$. $\mathcal{A}$ wins the game if $G(X_i \oplus Y_j) = y$. It should be clear that

$$\mathbf{Adv}^{\mathrm{Pre}}_{H(n)}(\mathcal{A}) \leq \mathbf{Adv}^{Game3}_{H(n)}(\mathcal{A})$$

From now on the analysis will be conducted with respect to *Game3*.

The notion of yield can also be used in preimage resistance analysis. Since the random experiment has changed we shall define the yield with respect to *Game3*.

Definition 6. *Let $S \subseteq \{0,1\}^{tn}$, then the yield of S is*

$$\mathrm{yield}(S) = \max_{\substack{C_1,\ldots,C_t \subseteq \{0,1\}^n \\ |C_1| = \cdots = |C_t| = q}} |S \cap (C_1 \times \cdots \times C_t)|$$

Let $S_W = \{X_i \oplus Y_j | 1 \leq i, j \leq tq\}$. Every time $\mathcal{A}$ makes a query, he/she will be able to evaluate G on some more elements in S_W. For $1 \leq i \leq tq$, let S_i be the set of strings in S_W which $\mathcal{A}$ can evaluate G on after the ith query (before the $(i+1)$th query), i.e. with respect to queries $\mathcal{A}$ have sent right after the ith query,

$$S_i = \{X_{i'} \oplus Y_{j'} | (G_k, X^{(k)}_{i'} \oplus Y^{(k)}_{j'}) \text{ is queried for all } 1 \leq k \leq t\}$$

Let $e_i = |S_i \backslash S_{i-1}|$, then right after $\mathcal{A}$ has made the ith query, he/she will be able to evaluate G on e_i more elements in S_W. Every new evaluation has probability 2^{-n} to match y, and $\sum_{i=1}^{tq} e_i$ is the total number of G evaluations $\mathcal{A}$ can compute.

Lemma 8. *For $e_1, \ldots, e_q$ as above,*

$$\sum_{i=1}^{tq} e_i \leq \mathrm{yield}(S_W)$$

Proof. Consider $\mathrm{yield}(S_W)$. By definition there exists a set $C_1, \ldots, C_t \subseteq \{0,1\}^n$, where $|C_1| = \cdots = |C_t| = q$, such that $|S_W \cap (C_1 \times \cdots \times C_t)|$ is maximized. If $\mathcal{A}'$ uses C_i as the set of queries to G_i, then

$$\sum_{i=1}^{tq} e_i = \mathrm{yield}(S_W)$$

On the other hand, if $\mathcal{A}'$ can query G_i such that $\sum_{i=1}^{tq} e_i > \mathrm{yield}(S_W)$, then by setting C_i to be the set of queries $\mathcal{A}'$ sent to G_i, $|S_W \cap (C_1 \times \cdots \times C_t)| > \mathrm{yield}(S_W)$, contradicting its definition. □

We can now upper bound the preimage-finding advantage of $\mathcal{A}$ with *Game3* and Lemma 8:

$$\begin{aligned}\mathbf{Adv}^{\mathrm{Pre}}_{H(n)}(\mathcal{A}) &\leq \mathbf{Adv}^{Game3}_{H_t(n)}(\mathcal{A}) \\ &\leq 2^{-n}\sum_{i=1}^{tq} e_i \\ &\leq 2^{-n}\,\mathrm{yield}(S_W) \qquad \text{by Lemma 8} \end{aligned} \tag{5.1}$$

Because $\mathrm{yield}(S_W) \leq |S_W| \leq (tq)^2$, the preimage resistance of H_t is at least $\Theta(2^{n/2})$.

Difficulties arise when we try to upper bound $\mathrm{yield}(S_W)$ using the same method as in Section 4.4. Firstly Lemma 3 stated by Shrimpton and Stam [17] cannot be used anymore because random functions no longer behave like random permutations. Secondly, the distribution of $X_i \oplus Y_j$ is uniform but not mutually independent. For $1 \leq i, i', j, j' \leq tq$,

$$X_i \oplus Y_j \oplus X_{i'} \oplus Y_j \oplus X_{i'} \oplus Y_{j'} \oplus X_i \oplus Y_{j'} = 0^n$$

Hence strings in S_W are at most 3-wise independent, making the task of bounding kcoll_{W_1} difficult for $k > 3$.

5.1 Tail Inequalities for Random Variables Under Exclusive-Or

The goal is to upper bound $\mathrm{yield}(S_W)$ for all possible adversaries, and we were able to reduce the problem further into an isolated mathematical problem, so that mathematicians can work on it without any cryptographic knowledge. Because non-adaptive adversaries are concerned as well, we will present two different problems, one for adaptive adversaries, and the other for non-adaptive ones.

Because both problems are similar and share the same notations, we shall list them here:

- Let $X_1, \ldots, X_{tq}$, $Y_1, \ldots, Y_{tq} \in \{0,1\}^{tn}$ be independent random variables with uniform distribution, where $t \geq 2$ is a constant positive integer.
- Let $Z_{ij} = X_i \oplus Y_j$ for $1 \leq i, j \leq tq$.
- Let $S = C_1 \times \cdots \times C_t$ be a product set, where $C_1, \ldots, C_t \subseteq \{0,1\}^n$ such that $|C_1| = \cdots = |C_t| = q$.

Problem (Adaptive Adversaries) The goal is to find non-trivial values B and k such that

$$\Pr\left[\exists S(|\{Z_{ij}|1 \leq i, j \leq tq\} \cap S| \geq k)\right] \leq B$$

The relationship between this problem and $\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A})$ is reflected by the following theorem.

Theorem 6. *Let functions k, B be a solution to the problem described above, and let $\mathcal{A}$ be an adaptive adversary, then*

$$\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}) \leq k2^{-n} + B$$

Proof. Note that $\{Z_{ij} | 1 \leq i, j \leq q\}$ has the same distribution as S_W.

$$\begin{aligned} \Pr\left[\exists S(|\{Z_{ij} | 1 \leq i, j \leq tq\} \cap S| \geq k)\right] &\leq B \\ \Pr\left[\exists S(|S_W \cap S|) \geq k\right] &\leq B \\ \Pr\left[\max_S |S_W \cap S| \geq k\right] &\leq B \\ \Pr\left[\mathrm{yield}(S_W) \geq k\right] &\leq B \end{aligned}$$

Together with inequality (5.1) we have

$$\begin{aligned} \mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}) &\leq \Pr\left[\mathrm{yield}(S_W) < k\right] k2^{-n} + \Pr\left[\mathrm{yield}(S_W) \geq k\right] 1 \\ &\leq k2^{-n} + B \end{aligned}$$

□

In case of non-adaptive adversaries, the problem is similar. The differences are that S is fixed and the adversary concerns about $|S_W \cap S|$ instead of $\mathrm{yield}(S_W)$, since the notion of yield does not make sense if the adversary is non-adaptive.

Problem (Non-Adaptive Adversaries) The goal is to find non-trivial values B and k such that for any fixed S as described in the list of notations above,

$$\Pr\left[|\{Z_{ij} | 1 \leq i, j \leq tq\} \cap S| \geq k\right] \leq B$$

The reduction theorem and proof are similar as well.

Theorem 7. *Let functions k, B be a solution to the problem described above, and let $\mathcal{A}^*$ be a non-adaptive adversary, then*

$$\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) \leq k2^{-n} + B$$

Proof. As shown above, $\{Z_{ij} | 1 \leq i, j \leq tq\}$ has the same distribution as S_W.

$$\begin{aligned} \Pr\left[|\{Z_{ij} | 1 \leq i, j \leq tq\} \cap S| \geq k\right] &\leq B \\ \Pr\left[|S_W \cap S| \geq k\right] &\leq B \end{aligned}$$

Let C_i be the set of queries which A^* send to G_i for $1 \leq i \leq t$. Set $S = C_1 \times \cdots \times C_t$, the by definition $S_{tq} = S_W \cap S$.

In the proof of (5.1) we have

$$\begin{aligned} \mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) &\leq 2^{-n} \sum_{i=1}^{q} e_i \\ &= 2^{-n} |S_{tq}| \end{aligned}$$

Making use of $\Pr[|S_W \cap S| \geq k] \leq B$ results the bound

$$\begin{aligned}\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) &\leq \Pr[|S_W \cap S| < k]\, k2^{-n} + \Pr[|S_W \cap S| \geq k]\, 1 \\ &\leq k2^{-n} + B\end{aligned}$$

□

Compared to the original setting with adversaries and queries, the reduced problem greatly simplifies the original problem. Not only it removes the adversary, the use of queries and the whole preimage finding task are abstracted away, and the most important of all, this problem accounts to developing a special tail inequality which can bound the preimage-finding advantage. Perhaps there are many instances of k and B which are solutions to either problems, but in order to give a tight bound we call for instances as small as possible.

5.2 Preimage Resistance Against Non-Adaptive Adversaries

In this section we will present a solution which upper bounds the preimage-finding advantage of non-adaptive adversaries. Moreover, we also have a lower bound of $|S \cap \{Z_{ij} | 1 \leq i, j \leq tq\}|$ for any fixed S, which together gives the preimage resistance of H_t against any non-adaptive adversary. An adaptive adversary can choose not to be adaptive, so the preimage resistance for adaptive adversaries is at most the same as that for non-adaptive adversaries. However, we conjectured that the preimage resistance for both types of adversaries are the same.

Theorems and proofs will be given in the perspective of the mathematical problem. Note that throughout this section $S = S_1 \times \cdots \times S_t$ is fixed with size q^t. Concrete query sets for a non-adaptive attack will be shown afterwards. The following proposition is crucial for the proofs:

Proposition 1. *(Chernoff Bound) Let $L_1, \ldots, L_r$ be mutually independent indicators which can each take a value of either 0 or 1. Let $L = \sum_{i=1}^{r} L_i$, then for any $\delta > 0$,*

$$\Pr[L < (1-\delta)\mathrm{E}[L]] < \exp(-\mathrm{E}[L]\delta^2/2)$$

Moreover, for any $\delta > 2e - 1$,

$$\Pr[L > (1+\delta)\mathrm{E}[L]] < 2^{-\mathrm{E}[L]\delta}$$

Theorem 8. *For any $\delta > 2e - 1$,*

$$\Pr\left[|S \cap \{Z_{ij} | 1 \leq i, j \leq tq\}| > t(1+\delta)\frac{q^{t+2}}{2^{tn}}\right] < tq2^{-\frac{q^{t+1}}{2^{tn}}t\delta}$$

Hence for any non-adaptive adversary $\mathcal{A}^*$,

$$\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) \leq t(1+\delta)\frac{q^{t+2}}{2^{(t+1)n}} + tq2^{-\frac{q^{t+1}}{2^{tn}}t\delta}$$

Proof. Fix j, then

$$\begin{aligned}\mathrm{E}[|S \cap \{Z_{ij}|1 \le i \le tq\}|] &= tq\frac{|S|}{2^{tn}} \\ &= t\frac{q^{t+1}}{2^{tn}}\end{aligned}$$

regardless of explicit contents of S. Define indicators $L_1, \ldots, L_{tq}$ where

$$L_i = \begin{cases} 1 & \text{if } Z_{ij} \in S, \\ 0 & \text{otherwise} \end{cases}$$

$L_1, \ldots, L_{tq}$ are then independent since $X_1, \ldots, X_{tq}$ are independent. Let $L = \sum_{i=1}^{tq} L_i$, then $L = |S \cap \{Z_{ij}|1 \le i \le tq\}|$ and $\mathrm{E}[L] = t\frac{q^{t+1}}{2^{tn}}$. By the Chernoff Bound we have

$$\begin{aligned}\Pr\left[L > t(1+\delta)\frac{q^{t+1}}{2^{tn}}\right] &< 2^{-\frac{q^{t+1}}{2^{tn}}t\delta} \\ \Pr\left[|S \cap \{Z_{ij}|1 \le i \le tq\}| > t(1+\delta)\frac{q^{t+1}}{2^{tn}}\right] &< 2^{-\frac{q^{t+1}}{2^{tn}}t\delta}\end{aligned}$$

By applying a union bound on $j = 1, \ldots, tq$, an upper bound of $|S \cap \{Z_{ij}|1 \le i, j \le tq\}|$ is obtained.

$$\begin{aligned}\Pr\left[\exists j|S \cap \{Z_{ij}|1 \le i \le tq\}| > t(1+\delta)\frac{q^{t+1}}{2^{tn}}\right] &< tq2^{-\frac{q^{t+1}}{2^{tn}}t\delta} \\ \Pr\left[|S \cap \{Z_{ij}|1 \le i, j \le tq\}| > t(1+\delta)\frac{q^{t+2}}{2^{tn}}\right] &< tq2^{-\frac{q^{t+1}}{2^{tn}}t\delta}\end{aligned}$$

□

The lower bound of $|S \cap \{Z_{ij}|1 \le i, j \le tq\}|$ can be proven in a similar way.

Theorem 9. *For any $\delta > 0$,*

$$\Pr\left[|S \cap \{Z_{ij}|1 \le i, j \le tq\}| < t(1-\delta)\frac{q^{t+2}}{2^{tn}}\right] < tq\exp\left(-\frac{q^{t+1}}{2^{tn}}\frac{t\delta^2}{2}\right)$$

Proof. Again we start by fixing j. By the other form of the Chernoff Bound,

$$\begin{aligned}\Pr\left[L < t(1-\delta)\frac{q^{t+1}}{2^{tn}}\right] &< \exp\left(-\frac{q^{t+1}}{2^{tn}}\frac{t\delta^2}{2}\right) \\ \Pr\left[|S \cap \{Z_{ij}|1 \le i \le tq\}| < t(1-\delta)\frac{q^{t+1}}{2^{tn}}\right] &< \exp\left(-\frac{q^{t+1}}{2^{tn}}\frac{t\delta^2}{2}\right)\end{aligned}$$

By applying a union bound on $j = 1, \ldots, tq$, a lower bound of $|S \cap \{Z_{ij}|1 \le i, j \le tq\}|$ is obtained.

$$\begin{aligned}\Pr\left[\exists j|S \cap \{Z_{ij}|1 \le i \le tq\}| < t(1-\delta)\frac{q^{t+1}}{2^{tn}}\right] &< tq\exp\left(-\frac{q^{t+1}}{2^{tn}}\frac{t\delta^2}{2}\right) \\ \Pr\left[|S \cap \{Z_{ij}|1 \le i, j \le tq\}| < t(1-\delta)\frac{q^{t+2}}{2^{tn}}\right] &< tq\exp\left(-\frac{q^{t+1}}{2^{tn}}\frac{t\delta^2}{2}\right)\end{aligned}$$

□

We can now apply Theorem 8 to find the preimage resistance of H_t against non-adaptive adversary $\mathcal{A}^*$.

Corollary 3. *For* $q = o\left(2^{\frac{t+1}{t+2}n}\right)$,

$$\lim_{n\to\infty} \mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) = 0$$

Moreover, if $q = k2^{\left(\frac{t+1}{t+2}-\epsilon\right)n}$ *where* k, ϵ *are positive constants,* $\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*)$ *is negligible.*

Proof. Suppose $q = o\left(2^{\frac{t+1}{t+2}n}\right)$, set δ such that $\delta\frac{q^{t+2}}{2^{(t+1)n}} = c$ for some constant $c > 0$, then

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) &\leq t(1+\delta)\frac{q^{t+2}}{2^{(t+1)n}} + tq2^{-\frac{q^{t+1}}{2^{tn}}t\delta} \\
&= t\frac{q^{t+2}}{2^{(t+1)n}} + tc + tq2^{-tc\frac{2^n}{q}} \\
&= o(1) + tc + 2^{\log_2(tq) - tc\frac{2^n}{q}} \\
\lim_{n\to\infty} \mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) &\leq tc
\end{aligned}$$

Since c is arbitrary, we have $\lim_{n\to\infty} \mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) = 0$.

If $q = k2^{\left(\frac{t+1}{t+2}-\epsilon\right)n}$,

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) &\leq t(1+\delta)\frac{q^{t+2}}{2^{(t+1)n}} + tq2^{-\frac{q^{t+1}}{2^{tn}}t\delta} \\
&= t(1+\delta)k^{t+2}2^{-(t+2)\epsilon n} + 2^{\log_2(tq) - k^{t+1}t\delta 2^{\left(\frac{1}{t+2}-(t+1)\epsilon\right)n}}
\end{aligned}$$

If $\frac{1}{t+2} - (t+1)\epsilon > 0$, setting δ to be a constant is enough to make the advantage negligible. Otherwise set $\delta = 2^{\left((t+2)\epsilon - \frac{1}{t+2}\right)n}$, then

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{pre}}_{H_t(n)}(\mathcal{A}^*) &\leq t(1+\delta)k^{t+2}2^{-(t+2)\epsilon n} + 2^{\log_2(tq) - k^{t+1}t\delta 2^{\left(\frac{1}{t+2}-(t+1)\epsilon\right)n}} \\
&= tk^{t+2}2^{-(t+2)\epsilon n} + tk^{t+2}2^{-\frac{1}{t+2}n} + 2^{\log_2(tq) - tk^{t+1}2^{\epsilon n}}
\end{aligned}$$

The advantage is now negligible. □

Corollary 4. *For* $q = k2^{\frac{t+1}{t+2}n}$ *where* $k > 0$ *is a constant,*

$$\lim_{n\to\infty} \mathbf{Adv}^{Game3}_{H_t(n)}(\mathcal{A}^*) \geq c - \frac{c^2}{2}$$

for some constant $0 < c \leq 1$.

Proof. Set δ such that $0 < k^{t+2}(1-\delta) \leq 1$ and let $c = tk^{t+2}(1-\delta)$. By Theorem 9 we have

$$\begin{aligned}\Pr\left[|S \cap \{Z_{ij}|1 \leq i,j \leq tq\}| < tk^{t+2}(1-\delta)2^n\right] &< tk2^{\frac{t+1}{t+2}n} \exp\left(-\frac{q^{t+1}}{2^{tn}}\frac{t\delta^2}{2}\right) \\ &\leq O(1)e^{\frac{t+1}{t+2}n} \exp\left(-k^{t+1}2^{\frac{1}{t+2}n}\right) \\ &= O(1)\exp\left(\frac{t+1}{t+2}n - k^{t+1}2^{\frac{1}{t+2}n}\right)\end{aligned}$$

Hence the probability is negligible. Since $S_{tq} = |S \cap \{Z_{ij}|1 \leq i,j \leq tq\}|$, $\mathcal{A}^*$ can evaluate G for $c2^n$ distinct inputs almost surely. Let those inputs be $x_1, \ldots, x_{c2^n}$, all distinct. For any $1 \leq i,j \leq c2^n$ with $i \neq j$, events $G(x_i) = y$ and $G(x_j) = y$ are independent since x_i, x_j are pairwisely independent. Therefore the following statements hold:

$$\Pr[G(x_i) = y] = 2^{-n} \text{ and } \Pr[G(x_i) = G(x_j) = y] = 2^{-2n}$$

By inclusion-exclusion principle,

$$\begin{aligned}\Pr\left[\bigcup_{i=1}^{c2^n} G(x_i) = y\right] &\geq \sum_{i=1}^{c2^n} \Pr[G(x_i) = y] - \sum_{i=1}^{c2^n}\sum_{j=1}^{i-1} \Pr[G(x_i)) = G(x_j) = y] \\ &= c2^n 2^{-n} - \frac{c2^n(c2^n-1)}{2}2^{-2n} \\ &\leq c - \frac{c^2}{2}2^{2n}2^{-2n} \\ &= c - \frac{c^2}{2}\end{aligned}$$

□

In the original random experiment, $\mathcal{A}^*$ can query F_i^L and F_i^R such that with high probability, the number of W evaluations he/she can make is $\Omega(q^2)$. Together with the result from Corollary 3 the preimage resistance of H_t against non-adaptive adversaries is $\Theta\left(2^{\frac{t+1}{t+2}n}\right)$.

For concrete query sets which can be used to mount the attack, one example is to use the set $\{1_{\text{bin}}, \ldots, q_{\text{bin}}\}$ for all random functions. In general, the sets of queries to F_i^L have to be the same for all $1 \leq i \leq t$. All queries to F_i^R have to the same as well, so that the number of W evaluations is maximized.

As a remark to the compression function designed by Shrimpton and Stam [17], Theorem 9 applies to their construction as well. In their case $t = 1$, and according to Theorem 9 non-adaptive adversaries can find a preimage using $O(2^{\frac{2}{3}n})$ queries, which coincides with their estimation of preimage resistance.

5.3 Potential Approaches

What remains open is whether H_t is secure against adaptive preimage-finding adversaries up to $\Omega\left(2^{\frac{t+1}{t+2}n}\right)$ queries. Several approaches were taken, and we

believe that some of the tricks can bypass the difficulties brought by dependence between elements in $\{Z_{ij}|1 \le i,j \le tq\}$.

If Random Variables are Independent If Z_{ij} were uniform and independent for all $1 \le i,j \le tq$, i.e. $\{Z_{ij}|1 \le i,j \le tq\}$ had the same distribution as $\{U_1,\ldots,U_{(tq)^2}\}$ where U_i are uniform and independent random variables, the preimage resistance would be $\Theta\left(2^{\frac{t+1}{t+2}n}\right)$, since by the Chernoff Bound

$$\Pr\left[|S \cap \{U_1,\ldots,U_{(tq)^2}\}| > (1+\delta)t^2\frac{q^{t+2}}{2^{tn}}\right] < 2^{-t^2\frac{q^{t+2}}{2^{tn}}\delta}$$

for any fixed S. Applying the union bound over all possible S gives

$$\begin{aligned}\Pr\left[\exists S\left(|S \cap \{U_1,\ldots,U_{(tq)^2}\}| > (1+\delta)t^2\frac{q^{t+2}}{2^{tn}}\right)\right] &< \binom{2^n}{q}^t 2^{-t^2\frac{q^{t+2}}{2^{tn}}\delta}\\ &\le 2^{tqn}2^{-t^2\frac{q^{t+2}}{2^{tn}}\delta}\\ &= 2^{tqn-t^2\frac{q^{t+2}}{2^{tn}}\delta}\end{aligned}$$

The lower bound is similar as well.

$$\Pr\left[|S \cap \{U_1,\ldots,U_{(tq)^2}\}| < (1-\delta)t^2\frac{q^{t+2}}{2^{tn}}\right] < \exp\left(-t^2\frac{q^{t+2}}{2^{tn}}\frac{\delta^2}{2}\right)$$

$$\begin{aligned}\Pr\left[\exists S\left(|S \cap \{U_1,\ldots,U_{(tq)^2}\}| < (1-\delta)t^2\frac{q^{t+2}}{2^{tn}}\right)\right] &< \binom{2^n}{q}^t \exp\left(-t^2\frac{q^{t+2}}{2^{tn}}\frac{\delta^2}{2}\right)\\ &\le 2^{tqn}\exp\left(-t^2\frac{q^{t+2}}{2^{tn}}\frac{\delta^2}{2}\right)\\ &\le \exp\left(tqn - t^2\frac{q^{t+2}}{2^{tn}}\frac{\delta^2}{2}\right)\end{aligned}$$

Currently our bound for $|S \cap \{Z_{ij}|1 \le i,j \le tq\}|$ is not small enough to apply the union bound over all possible S.

Closed Product Sets Instead of studying $|S \cap \{Z_{ij}|1 \le i,j \le tq\}|$ for product sets S in general, product sets which are closed under the tertiary bitwise exclusive-or operation have consistent properties, defined formally as follows:

Definition 7. *A set S is closed under the tertiary bitwise exclusive-or operation if for any $s_1,s_2,s_3 \in S$,*

$$s_1 \oplus s_2 \oplus s_3 \in S$$

The product set $S = \{1_{\text{bin}},\ldots,q_{\text{bin}}\}^t$ is closed under tertiary xor. Such sets behave consistently with respect to $\{Z_{ij}|1 \le i,j \le q\}$ because whenever $Z_{ij}, Z_{i'j}, Z_{ij'}$ are in S, $Z_{i'j'} \in S$ with certainty.

There are many closed product sets. Based on the observation that $S = S_1 \times \cdots \times S_t$ is closed if and only if S_i are closed for all i, one can concentrate on finding

subsets of $\{0,1\}^n$ which are closed. Sets with elements sharing the same prefix are closed. i.e. for p a fixed string for length c, $\{p\|s|s \in \{0,1\}^{n-c}\}$ is closed. Moreover, let P be a permutation of bits, then for any closed set S_i, $\{P(s_i)|s_i \in S_i\}$ is closed. However, closed sets generated by these two methods do not include other closed sets like $\{1111, 1110, 1101, 1110, 0000, 0001, 0010, 0011\}$.

One can make use of these closed sets to study the behavior of general product sets under such distribution of Z_{ij}, as any product set can be partitioned into a set of closed sets. It should be noted that the closeness property may lead to a misconception that closed sets tend to give a larger intersection. In fact, closeness comes with a tradeoff. Suppose S is closed. If $Z_{ij}, Z_{i'j}$ are in S but not $Z_{ij'}$, then $Z_{i'j'} \notin S$ for sure. This leads us to a belief that no particular product set is more likely to produce either a larger or smaller intersection than any other product set of the same size.

Bipartite Graph Model Another approach is to formulate the mathematical problem as a graph problem. Given S, define bipartite random graph $G_S = (U, V, E_S)$ as follows:

- $U = \{u_1, \ldots, u_{tq}\}$.
- $V = \{v_1, \ldots, v_{tq}\}$.
- Edge $\{u_i, v_j\} \in E_S$ if and only if $Z_{ij} \in S$.

By definition, G_S is bipartite, and $|S \cap \{Z_{ij}|1 \leq i, j \leq tq\}| = |E_S|$. An advantage of such formulation is, any fixed forest of size k occurs with probability $\left(\frac{|S|}{2^{tn}}\right)^k$, regardless of the contents of S. If S is closed under tertiary xor, G_S is composed by a set of complete bipartite components, since $\{u_i, v_j\}, \{v_j, u_{i'}\}, \{u_{i'}, v_{j'}\} \in E_S$ implies that $\{u_i, v_{j'}\} \in E_S$. In general, note that any fixed component with exactly k vertices occurs with probability $\left(\frac{|S|}{2^{tn}}\right)^k$, and can at most contain k^2 edges, so one might be able to give a bound on $|E_S|$ by bounding the probability of the size of the largest component as well as bounding the total number of components in G_S.

A Motivating Example The following example is a comparison between a closed set and an unclosed set, illustrating the fact that closed sets do not give strictly larger intersections. Although the parameters we give do not fit into the random experiments we defined previously, they are set on purpose to keep the example small and simple.

Example: Let $n = 4$. Let $S = \{0000, 0001, 0010, 0011\}$ and $S' = \{1100, 0101, 0011, 1000\}$. Consider G_S and $G_{S'}$ for $q = 2$ and $t = 1$, where $\{Z_{ij}|1 \leq i, j \leq q\} = \{Z_{11}, Z_{12}, Z_{21}, Z_{22}\}$ and

$$Z_{11} \oplus Z_{12} \oplus Z_{21} = Z_{22}$$

There are four immediate observations:

1. $\Pr[Z_{ij} \in S] = \Pr[Z_{ij} \in S'] = \frac{1}{4}$ for any i, j, so $\mathrm{E}[|E_S|] = \mathrm{E}[|E_{S'}|]$.
2. S is closed under tertiary xor.
3. For any distinct $s'_1, s'_2, s'_3 \in S'$, $s'_1 \oplus s'_2 \oplus s'_3 \notin S'$.
4. $|E_S|$ can never be 3.

Let $p = \frac{1}{4}$. Stated as a property of closed sets, $\Pr[|E_S| = 4] = p^3$. Consider $\Pr[|E_{S'}| = 4]$. By observation $|E_{S'}| = 4$ if and only if $Z_{11}, Z_{12}, Z_{21} \in S'$ and are not distinct. Given $Z_{11}, Z_{12}, Z_{21} \in S'$ there are 40 out of 64 cases where they are not distinct, thus

$$\Pr[|E_{S'}| = 4] = \frac{5}{8}p^3 < p^3 = \Pr[|E_S| = 4]$$

as expected.

Now consider $\Pr[|E_{S'}| = 3]$. There are a total of 4 graphs possible. Since they occur with the same probability, fix a particular graph. A path of length 3 exists with probability p^3. Given 3 edges exist, the last edge does not exist with probability $1 - \frac{5}{8} = \frac{3}{8}$, so

$$\Pr[|E_{S'}| = 3] = \frac{3}{2}p^3 > 0 = \Pr[|E_S| = 3]$$

There are a total of 6 graphs possible which have exactly 2 edges. Fix the graph with edges $\{u_2, v_1\}, \{u_2, v_2\}$ missing. Edges $\{u_1, v_1\}, \{u_1, v_2\}$ exists with probability p^2, and $\{u_2, v_1\}$ does not exist with probability $1 - p$. For S, since $Z_{22} \notin S$ if Z_{11}, Z_{22} are but not Z_{21}, $\Pr[|E_S| = 2] = 6p^2(1 - p)$.

For S', imagine the situation as $\{u_1, v_1\}$ appearing and $\{u_2, v_1\}, \{u_2, v_2\}$ missing, which occurs with probability $p(1 - p)^2$, then $\{u_1, v_2\}$ appears with probability $\frac{7}{24}$. Hence

$$\Pr[|E_{S'}| = 2] = \frac{21}{16}p(1 - p) < \frac{3}{2}p(1 - p) = \Pr[|E_S| = 2]$$

There are a total of 4 graphs possible which have exactly 1 edge. Fix the graph with only edge $\{u_1, v_1\}$ appearing. The probability that $\{u_1, v_1\}$ appears but not $\{u_1, v_2\}, \{u_2, v_1\}$ is $p(1 - p)^2$, both for G_S and $G_{S'}$. For S, given only $\{u_1, v_1\}$ appears, $\{u_2, v_2\}$ does not appear with probability $\frac{2}{3}$. For S', given only $\{u_1, v_1\}$ appears, $\{u_2, v_2\}$ does not appear with probability $1 - \frac{7}{24} = \frac{17}{24}$.

$$\Pr[|E_{S'}| = 1] = \frac{17}{6}p(1 - p)^2 > \frac{8}{3}p(1 - p)^2 = \Pr[|E_S| = 1]$$

The steps of showing

$$\Pr[|E_{S'}| = 0] = \frac{17}{72}(1 - p)^3 < \frac{7}{9}(1 - p)^3 = \Pr[|E_S| = 0]$$

is tedious and will not be shown here.

The example above shows not only that $\mathrm{E}[E_S] = \mathrm{E}[E_{S'}]$, but the distribution of $|E_S|$ and $|E_{S'}|$ are somewhat concentrated around the expected value. The value of Z_{22} is determined by its 3 counterparts: Z_{11}, Z_{12} and Z_{21}, and it seems that given most of its counterparts not in S or S', the probability of Z_{22} falling into S or S' is closer to being independent. This might be the explanation of why the distribution of $|S \cap \{Z_{ij} | 1 \leq i, j \leq q\}|$ is similar to uniform in general. If n is large and $q \ll n$, counterparts of Z_{ij} fall into S with a relatively low probability, so Z_{ij} might fall into S with a probability close to being uniformly independent.

Chapter 6

Conclusion

In this paper we have shown a class of variant constructions from the Benes construction using length-preserving public random functions, driven by parameters n and t. Besides having $t = 1$ as a degenerate case, construction H_t has a collision resistance of $\Theta(2^{n/2})$ for $t \geq 2$. Moreover, the preimage resistance of H_t against non-adaptive adversaries is $\Theta(2^{\frac{t+1}{t+2}n})$ queries, and we conjecture that this is indeed the case for *all* adversaries. However, the work is far from done. The preimage resistance for adaptive adversaries has not yet been proven, and currently no attacks better than the one we gave in Section 5.2 is found. To facilitate the analysis we reduced the preimage finding task into an isolated mathematical problem, and offered possible approaches to solve it.

Our research leads to a general mathematical question about exclusive-or. Although the bitwise exclusive-or operation is commonly used in computer applications, we realize that we actually do not understand its distribution when the operands are random. Shrimpton and Stam [17] conjectured that the number of k-way collisions for the distribution $A \oplus B$, where A and B are independent and uniform distributions, is asymtotically a Poisson distribution, and they support the statement with experimental data. If this is really the case then it will contribute greatly to the design and analyses of cryptographic schemes.

Currently, in terms of collision and preimage resistances, **eSSt** (See Section 2.3.1) have both of them optimal using three layers. Our two-layered construction has optimal collision resistance, and we believe that it can have a preimage resistance arbitrarily close to being optimal. Other properties, multicollision resistance or indifferentiability for example, are worthy to be investigated as well. When additional properties are concerned, a research area in hash function designs will be investigating butterfly transformations. The Benes construction is just a double butterfly transformation, and the behavior of a k-round butterfly network remains an open problem. It bears a number of similarities with the Feistel transformation. While Feistel networks are popular, little research has been done on butterfly transformations. One can also bring it to the ideal cipher model, analyzing its behavior when the primitives are random permutations, or even combining it with Feistel transformations.

A more practical problem will be finding a suitable candidate replacing the public random function. Since a public random function is not realizable, otherwise a random oracle is also realizable, finding a replacement such that the construction is still secure is non-trivial. In fact, identifying properties needed for replacements is already a problem by itself.

We will continue our analysis on preimage resistance, and perhaps conduct analyses on multicollision resistance as well as indifferentiability.

Bibliography

[1] William Aiello and Ramarathnam Venkatesan. Foiling birthday attacks in length-doubling transformations - Benes: A non-reversible alternative to Feistel. In *Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 307–320, 1996.

[2] Elena Andreeva, Gregory Neven, Bart Preneel, and Thomas Shrimpton. Seven-property-preserving iterated hashing: ROX. In *Advances in Cryptology — ASIACRYPT 2007*, pages 130–146, 2007.

[3] Elena Andreeva, Gregory Neven, Bart Preneel, and Thomas Shrimpton. Three-property preserving iterations of keyless compression functions. In *ECRYPT Hash Workshop 2007*, 2007.

[4] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *CCS '93: Proceedings of the 1st ACM conference on Computer and Communications Security*, pages 62–73. ACM Press, 1993.

[5] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption - how to encrypt with RSA. In *Advances in Cryptology — EUROCRYPT '94, LNCS 950*, pages 92–111. Springer-Verlag, 1995.

[6] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. pages 399–416. Springer-Verlag, 1996.

[7] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, 2004.

[8] Jean-Sebastien Coron, Yevgeniy Dodis, Cecile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. pages 430–448. Springer-Verlag, 2005.

[9] Jean-Sebastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. Cryptology ePrint Archive, Report 2008/246, 2008.

[10] Ivan Damgård. A design principle for hash functions. In *Advances in Cryptology — CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer-Verlag, 1989.

[11] Ueli Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *Theory of Cryptography Conference — TCC 2004*, volume 3378 of *Lecture Notes in Computer Science*, pages 21–39. Springer-Verlag, February 2004.

[12] Ueli Maurer and Stefano Tessaro. Domain extension of public random functions: Beyond the birthday barrier. In *Advances in Cryptology — CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 187–204. Springer-Verlag, 2007. Full version available from http://eprint.iacr.org/2007/229.

[13] Jacques Patarin. A proof of security in $O(2^n)$ for the Benes scheme. In *AFRICACRYPT*, pages 209–220, 2008.

[14] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In *Fast Software Encryption 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388, 2004.

[15] Phillip Rogaway and John P. Steinberger. Constructing cryptographic hash functions from fixed-key blockciphers. In *Advances in Cryptology — CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 433–450. Springer, 2008.

[16] Phillip Rogaway and John P. Steinberger. Security/efficiency tradeoffs for permutation-based hashing. In *Advances in Cryptology — EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 220–236. Springer, 2008.

[17] Thomas Shrimpton and Martijn Stam. Building a collision-resistant compression function from non-compressing primitives. In *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 643–654. Springer, 2008.

[18] Martijn Stam. Beyond uniformity: Better security/efficiency tradeoffs for compression functions. In *Advances in Cryptology — CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 397–412. Springer, 2008.

[19] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In *Advances in Cryptology — CRYPTO 2005*, volume 3621 of *LNCS*, pages 17–36. Springer, 2005.

[20] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In *Advances in Cryptology — EUROCRYPT 2005*, pages 19–35, 2005.