The concrete security of practical cryptographic constructions - A survey

Nikolas Melissaris*

Rutgers University

Abstract. Information-theoretic cryptography argues about security against unbounded adversaries. We discuss provable-security results for block ciphers based both on the Feistel and the Substitution Permutation paradigm. We survey important results for the concrete security of practical instantiations of cryptographic primitives and constructions based on them.

1 Introduction

One can partition cryptographic research into two classes on the basis of the assumptions made during any proofs of security: classical and concrete cryptography. So-called "classical" cryptography is founded on the relatively mild assumption that one way functions exist, continuing on to develop primitives which achieve a multitude of goals: encryption, authentication, integrity, commitment schemes, digital signatures and so on. Classically, security is defined as resilience against adversaries with limited resources, a natural assumption we place on an adversary. In contrast, the less studied "concrete" cryptography considers security against unbounded adversaries. This distinction is highly important — classical cryptography first assumes the existence of a primitive, namely the one way function, whereas concrete cryptography posits the existence of specific objects with certain desirable combinatorial properties and ultimately proves unconditional security. These two approaches will be addressed with greater detail in Sections 1.1 and 1.2.

Roadmap. For the remainder of the introduction, we will define the two "competing" approaches to cryptography. In Section 2 we introduce the necessary preliminaries that establish the proper background. In Section 3 we discuss practical constructions of private-key primitives. In Section 4 we introduce and present the most important and recent *indistinguishability* results while in Section 5 we argue about the necessity of the stronger notion of *indifferentiability* and present the most recent and important results within framework. In Section 6 we present some interesting open problems that we identified while surveying this area.

* The research area that was surveyed in this work is highly complicated and technical. Most of the papers that were presented have full versions that are over fifty pages and their main contributions are proven through series of lemmas that span at least ten pages. I would like to thank John Steinberger, Shan Chen, and Yuanxi Dai for their guidance through this work. I am grateful to them for patiently explaining (and pointing out!) to me the subtle differences of the involved combinatorial arguments that are presented. I would also like to thank Kelsey Horan and Hafiz Asif for their insightful comments and suggestions. Finally, I'm grateful to Periklis Papakonstantinou for his constructive criticism of this manuscript.

1.1 Cryptography: the complexity theoretic approach (the asymptotic approach)

We will now define the asymptotic (also known as *complexity theoretic*) approach to arguing the security of cryptographic schemes. Complexity theoretically, cryptographic security is proven by first assuming that certain problems are hard for adversaries with limited computational resources [1]. Those hardness assumptions, assumptions on the intrinsic computational complexity of particular problems, are studied extensively throughout *complexity theory*. Security proofs are then done via reduction; a cryptographer essentially shows that if an adversary manages to break the cryptosystem under consideration then he can solve a (hard) problem considered to be infeasible. In this asymptotic approach a cryptographic scheme is considered secure if *every* probabilistic, polynomial time (PPT) adversary has only a negligible probability of breaking the scheme.

As a concrete example consider the following: we assume that factoring an integer, which is a product of two primes, is a problem that has no efficient solution[2]. The best method developed for factoring is barely much more efficient than brute-force, i.e. dividing the integer by all possible prime factors. The RSA cryptosystem was built on the assumption that factoring is hard. Then, it was proven that *efficiently* breaking RSA would imply that the adversary can provide a legendary contribution to mathematics, an algorithm to *efficiently* factor a product of two primes. Although the hardness of factoring is unproven, most of the world's secure communication depends on this assumption.

The shortcomings of complexity theoretic cryptography are not only visible up close, the entire field rests upon the famous $\mathcal{P}$ vs $\mathcal{NP}$ problem. If $\mathcal{P} \neq \mathcal{NP}$ and one way functions[3] exist, then one can derive most private-key cryptography [IL89]. Alternatively, if $\mathcal{P} = \mathcal{NP}$ (quite unlikely, but still open) then one way functions do not exist and all cryptographic constructions with complexity theoretic security are broken.

For a quick, informal proof, consider the following: Let $y = f(x)$ the output of a one way function f. Define L to be the language of pairs (x', y) such that x' is a prefix of some x for which $f(x) = y$. L's membership in $\mathcal{NP}$ is straightforward since f is polynomial time computable and x itself can serve as a witness. We can then use a decider D for L to invert f (remember that we have assumed $\mathcal{P} = \mathcal{NP}$). Upon receiving $(y, 1^n)$ for some security parameter n, starting by the empty string ϵ, use D to add bits to the prefix until we have (x, y). Since we know that a preimage of y exists and it has length at most n, then our algorithm runs in polynomial time.

In some sense there is a need for something "stronger"; ideally, one would like to establish security independently of any unproven assumptions.

[1] Typically, we consider polynomially bounded adversaries.

[2] We have no proof of this but we believe that centuries of futile efforts by very smart people is a good indicator.

[3] A function f is a one way function (OWF) if it's efficiently computable and a polynomial-time algorithm is able to invert f with *negligible* probability when the input to the function is chosen uniformly at random.

1.2 Cryptography: the information-theoretic approach (the concrete approach)

The concrete approach requires no assumptions on the power of an adversary, allowing for security against an all powerful attacker. In this case the security definition references measurements typically studied in *information theory*, such as entropy or statistical distance. Therefore, this area is also called *information-theoretic cryptography* but herein is referred to as concrete cryptography. Specifically, security is measured by bounding the success probability of *any* adversary spending some specific amount of computational effort. The phrase computational effort does not refer to time, as time is no object to an unbounded adversary, but instead refers to the total number of queries the adversary issues to an algorithm called an oracle. Concretely, security does not depend on a specific computational model, thereby ensuring that these schemes cannot be compromised even in the face of new computational technology or paradigms, such as the development of quantum computers.

In addition to the security notions above, concrete instantiations of cryptographic primitives tend to be much more efficient that the theoretic ones. For example, the best pseudorandom function due to Naor and Reingold [NR04] which is based on number theoretic assumptions, has key length quadratic to the length. In contrast, typical block ciphers have key length about the size of the input.

This is the area that we will survey in this work. Research in this direction is deeply mathematical, based entirely on constructions and proofs. Although an overview of a large portion of the relevant literature is provided, a selection of papers will be highlighted as we believe they are of great importance to the field. Simple but important proofs will be presented in full, while lengthy, involved proofs will be presented at a high level, and any complicated calculations will be omitted for compactness.

2 Preliminaries and Notation

This section presents notions and constructions relevant to this text. Prior to surveying the area of information-theoretic cryptography, it must be placed within the larger map by taking a step back and analyzing the big picture of cryptographic research.

The introduction mentioned that private-key cryptography can only be developed from the (mild) assumption that one way functions (OWF) exist, a statement that is not very intuitive. How exactly does a cryptographer construct a cryptographic scheme starting from an OWF? The answer was provided in a sequence of very important results spanning a e decade, showing that from any OWF one can construct a pseudorandom generator (PRG), from which a pseudorandom function (PRF) can be built, and finally, a pseudorandom permutation (PRP) (essentially, in our setting, a block cipher). Below is an overview and a description of these constructions:

$$\text{OWF} \xrightarrow{[\text{HILL99}]} \text{PRG} \xrightarrow{[\text{GGM84}]} \text{PRF} \xrightarrow{[\text{LR88}]} \text{PRP}$$

Pseudorandom Permutations. Intuitively, a PRP is a permutation that looks like a completely random permutation to any efficient adversary. To be more precise, the permutation

appears to be selected uniformly at random from the space of all possible permutations on n-bits, denoted P_n. It follows that the size of P_n is $(2^n)!$. In fact, to make the security requirement stronger, the adversary is allowed access to the inverse of the permutation. Formally:

Definition 1. *Let $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be an efficient, length preserving, keyed function. We call F a strong pseudorandom permutation if for all PPT distinguishers D, there is a negligible function* negl *such that:*

$$\left|\Pr[D^{F_k(\cdot),F_k^{-1}(\cdot)}(1^n)] = 1 - \Pr[D^{f(\cdot),f^{-1}(\cdot)}(1^n) = 1]\right| \leq \text{negl}(n)$$

where the first probability is taken over the uniform selection of $k \in \{0,1\}^n$ and the second probability is taken over the uniform choice of f from the space of all permutations on n bits.

As this survey discusses block ciphers, the real world instantiations of pseudorandom permutations (PRP), the PRP is the theoretical building block of the survey. From PRPs one can construct (almost) all other private-key primitives, such as private-key encryption, message authentication schemes, and many others. The connection between this primitive and block ciphers is explained below.

Block Ciphers. As mentioned, block ciphers are real world instantiations of PRPs. Block ciphers can simply be seen as keyed permutations $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ where for all keys k, the map $E(k,\cdot)$ must be efficiently invertible, $E^{-1}(k,\cdot)$.

In one of the most important works, also cited above, Luby and Rackoff [LR88] created a PRP from a psuedorandom function (PRF). Thus, it is important to define the PRF as well. For formal treatment of the Luby-Rackoff construction, see Section 5.1.

Pseudorandom Functions. Since the space of permutations is a subset of the space of all functions the set of PRFs is a superset of the set of PRPs. In the same fashion as above, a PRF must look random just as a PRP looks random to any polynomial time distinguisher. Informally, a distinguisher's task is to determine if a given function is selected uniformly at random from the space of all functions (size 2^{n2^n}), or is instead selected from a keyed family of functions, i.e. F_k for some uniform k.

Definition 2. *Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length preserving, keyed function. We call F a pseudorandom function if for all PPT distinguishers D, there is a negligible function* negl *such that:*

$$\left|\Pr[D^{F_k(\cdot)}(1^n)] = 1 - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \leq \text{negl}(n)$$

where the first probability is taken over the uniform selection of $k \in \{0,1\}^n$ and the second probability is taken over the uniform choice of f from the space of all functions on n bits.

We gave the known definitions of PRFs and PRPs where the adversary is computationally bounded, but in this survey the actual proofs themselves look into the information-theoretic case where the difference of the probabilities is maximized over all distinguishers, only bounded by the number of queries to the functions and permutations, but with no further restrictions on their time complexities.

While the limitations placed on the adversaries are described thoroughly, the general notion of security remains undefined in this text. The notion of cryptographic security can be demystified by presenting security definitions and particular *attack models*.

Definitions of Security. The algorithms for encryption/decryption, E, D, are modeled as black boxes, *oracles*, O^E and O^D that the adversary has access to. Some of the most studied threat models are game-based security definitions, outlining a game that the adversary plays with access to only O^E or both O^E and O^D. This distinction plays a big part in the number of rounds that make the Luby-Rackoff construction secure in Section 5.1. The formal definitions follow:

- *Chosen-plaintext attack (CPA).* In this model, an adversary can ask O^E to encrypt plaintexts of his choice. After some time the adversary must produce two messages m_0 and m_1 to submit to the oracle. The oracle encrypts one of the two messages and returns $E(m_b)$ to the adversary where $b \in \{0, 1\}$ is selected uniformly at random. If the adversary can detect with probability non-negligibly over $1/2$ which message was encrypted, the adversary wins the game.

- *Chosen-ciphertext attack (CCA).* In this model, an adversary has access to O^D as well. Again, after requesting encryptions (resp. decryptions) to plaintexts (resp. ciphertexts) of his choice the adversary produces two messages m_0 and m_1. The oracle flips a coin, encrypts one of the messages, and returns $E(m_b)$ to the adversary. If the adversary can tell with probability over $1/2$ which message was encrypted then the adversary wins the game.

  Example. Here is a very simple example illustrating that CCA security is a stronger notion than CPA security. Below is an encryption scheme which achieves CPA-secure but fails to achieve CCA security. Define the following encyption scheme:

  $$E(k, x) = (r, F_k(r) \oplus x))$$

  where $r \in \{0, 1\}^n$ is selected uniformly at random and F_k is a pseudorandom function. The attacker will send $m_0 = 0^n$ and $m_1 = 1^n$ to the encryption oracle and receives $c = E_k(m_b)$. Now, he flips the first bit of the ciphertext creating $c' \neq c$ and submits it to the decryption oracle. Now the plaintext returned by the oracle is just m_0 or m_1 with the first bit flipped. So, with probability 1 the adversary can win the CCA game.

Usually we have an attacker that issues specific queries to the construction. To prove the computational security in such a case, we follow two steps:

1. We model the PRF (resp. PRP) as a truly random function (resp. permutation) and prove the security unconditionally.

2. We observe that when the random function (resp. permutation) is replaced by a PRF (resp. PRP) then any efficient adversary that succeeds, can be converted into an efficient distinguisheer for the pseudorandom primitives. This contradicts the itractability assumption.

In the Real World. Although such schemes are theoretically secure, one remaining question is if the constructions have any use in the real world — can these primitives actually be constructed in a useful manner?

3 Practical Constructions of Private-Key Primitives

Block ciphers are typically constructed using one of the following two paradigms: Feistel networks [Fei73] or substitution-permutation networks (SPNs) [Sha49, Fei73]. Both constructions are surveyed in this work.

Feistel Networks. The following is an r-round Feistel construction Ψ_r, which implements a permutation on $2n$-bits: let $r \geq 0$ and let $F_1, \ldots, F_r : \{0,1\}^n \to \{0,1\}^n$. A $2n$-bit input is parsed as two halves (L_0, R_0) and the i-th round is computed as:

$$L_i = R_{i-1} \quad , \quad R_i = L_{i-1} \oplus F_i(R_{i-1})$$

and the output is (L_r, R_r) after r-rounds. A simple example that uses three rounds can be seen in Figure 1.

An important aspect of the Feistel construction is that the underlying function F is not required to be invertible. Therefore, the Feistel network is a way to build an invertible function from smaller, non-invertible, functions. It should also be mentioned that Luby-Rackoff constructed a pseudorandom permutation starting from pseudorandom functions in [LR88] using a Feistel construction, the details of which are analyzed in Section 5.1.

Historically, the Feistel construction was very important as it was used for the data encryption standard (DES) in 1977 [Nat99]. DES was subsequently replaced in 2001 by the advanced encryption standard (AES) [Nat01], a cryptosystem that is still used in virtually all communications. In contrast, AES is built on the *substitution permutation* paradigm, addressed in the following section.

Substitution Permutation Networks. Substitution permutation networks (SPN) (cf. Figure 2) are a special case of the confusion-diffusion paradigm first introduced by Shannon [Sha49]; the idea of an SPN is to construct a random-looking permutation F from many smaller random permutations, f_i.

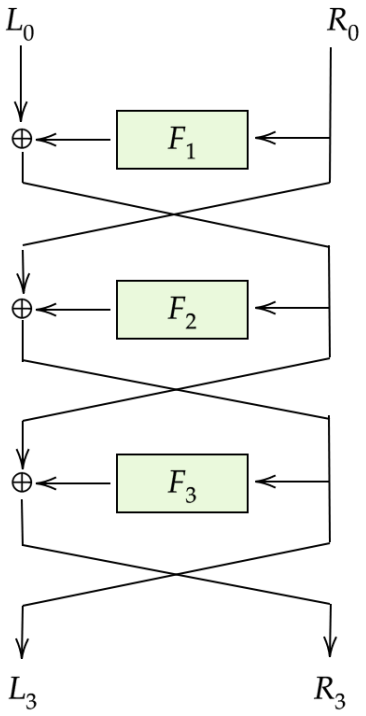

Fig. 1. A 3-round Feistel construction. We will see this later when we discuss about the famous result of [LR88].

> **Example.** Assume that you want to construct an F with a block-length of 64 bits. Selecting a key for F will specify 8 smaller permutations $f_1, \ldots, f_8$ that each have a block of 8 bits. Then, split the 64 bit input into 8[4] parts and define $F(x) = f_1(x_1)\| \ldots \| f_8(x_8)$. This is called the *confusion* step. It should be clear that F is not pseudorandom defined in this way[5], therefore a *diffusion* step, where the bits of the output are permuted, is required.

An SPN starts with one or more public permutations on n-bits that "look random" (called S-boxes) and extend these permutations to construct a keyed pseudorandom permutation on wn-bit inputs for some w, by applying some number of iterations of the following steps:

- Substitution step: split the wn-bit state into w n-bit blocks, then apply an S-box to each n-bit block.

[4] We define the input: $x = x_1||\ldots||x_8$

[5] If x and x' only have their first bit different then only $f_1(x)$ and $f_1(x')$ would be different and $F(x)$ and $F(x')$ would only differ in their first part. If F were a truly random permutation then changing the first bit of the input would be expected to affect all the chunks of the output.

- Permutation step: pass the entire wn-bit state through a non-cryptographic, keyed permutation.

SPNs can be further characterized as linear (or non-linear) based on whether the permutation step is a linear (or non-linear) function.

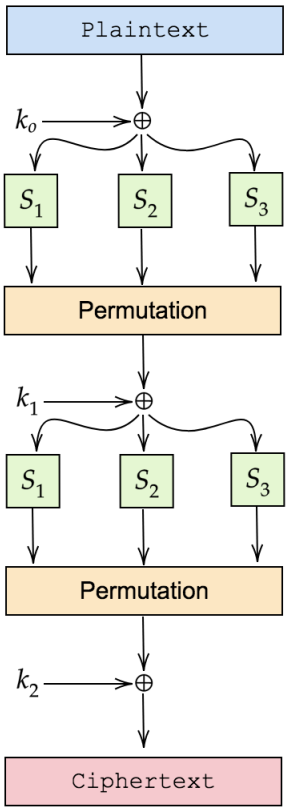

Fig. 2. The three keys k_0, k_1, k_2, are derived from a single key. This series of keys is called a key schedule and the derived keys are called round keys.

Along with the construction of practical schemes comes the need to "measure" their behavior and the proximity of the construction to it's theoretical counterpart. In particular, can such a construction can be distinguished from a PRP by any adversary? The concept of *indistinguishability* is described in the following section.

4 Indistinguishability

This section describes a concept to is used towards the analysis of cryptographic constructions, namely indistinguishability. Secure encryption schemes are created from small[6] building blocks

[6] If we wanted to have a truly random function on n-bits we would need to specify the output for every possible input. There are 2^n possible inputs, each needing $\log(n)$ bits to be described so we would $2^n \log(n)$ bits only to specify the function.

(called *primitives*) such as pseudorandom generators, pseudorandom permutations, and hash functions. Naturally, it would be ideal to have a method to construct these primitives for use in practice while preserving any nice properties, in order to establish security of the designed scheme. Unfortunately, this is not an easy task.

4.1 The Ideal Cipher and the Random Oracle Model

One approach is to rely on the so-called *ideal primitive model*. In this model, the building blocks of a cryptographic scheme are replaced with an idealized information-theoretic version. The most prominent example of this idea is the Random Oracle Model [BR93]. In the random oracle model, instead of making the assumption that a scheme is secure when using a specific hash function (e.g SHA-256[7]), one assumes access to a truly random function. As the notion of hash functions has not yet been introduced, it is important to do so now for this example.

Hash Functions. Generally speaking, hash functions are functions that take inputs of (possibly different) lengths and compress them into a shorter, fixed length output. In cryptography, hash functions are designed to be collision resistant, meaning that an adversary with access to the hash function cannot efficiently produce inputs x, y such that $x \neq y$ but $H(x) = H(y)$. *Families* of hash functions $H_k(x)$ are such that for a uniformly selected key k, it is hard for an efficient adversary to find x, y such that $x \neq y$ but $H_k(x) = H_k(y)$ even after interacting with the function for some polynomially bounded time. It is important to note that even though hash functions are deterministic their output size makes it infeasible to invert them as they model random functions.

The example above references SHA-256, a specific hash function where the output is 256 bits which means that there are 2^{256} possible outputs. This implies that -by the birthday paradox- an adversary needs about 2^{128} hashes before there is a significant probability of finding a collision. This is a *very large number*. With the ability to compute 10^{15} hashes per second, it would still take about 10^{13} years to gather as many hashes. In comparison, the universe is about 10^9 years old.

In the Ideal Cipher model it is assumed that all parties have access to a random permutation as well as its inverse. For example, as opposed to performing difficult calculations based on how AES is constructed it is assumed that AES will behave as an ideal cipher. This modelling assumption is not unfounded; years of scientific effort have been devoted towards the cryptanalysis of AES since its inception without any significant progress in terms of breaking it [FKL+00, GM00, HLL+00, Luc00, Bir04].

Security is then proven by assuming an unbounded adversary that has oracle access to the idealized primitive. The ideal primitive model has been used extensively and for good reason – this model helps with the design of simple, practical and efficient solutions to numerous problems.

The next section formally discusses how to argue and prove that a given construction is close to an idealized one.

[7] Although the SHA family of functions are *keyless*, we can think of SHA as being a keyed hash function by thinking of the key as being part of the input (appropriately embedded)

4.2 Indistinguishability

Above, we alluded to the notion of "closeness" between two constructions: the real construction and the idealized one. These two constructions should be *indistinguishable* by any unbounded adversary that has oracle access to both the real construction and the idealized version[8]. Formally:

Definition 3. *Two systems $\mathcal{S}$ and $\mathcal{T}$ are indistinguishable if there exists a negligible function* negl *such that for the advantage of all distinguishers D and for all n we have:*

$$|\Pr[D(\mathcal{S}) = 1] - \Pr[D(\mathcal{T}) = 1]| \leq \text{negl}(n)$$

The following section surveys some existing indistinguishability results that were proved for Feistel constructions.

4.3 Indistinguishability of Feistel constructions

Feistel networks have received great attention by the research community, resulting in a line of important results starting with Luby and Rackoff [LR88]. As mentioned, Luby-Rackoff proved that it is possible to construct pseudorandom permutations (which can be used as block ciphers) from pseudorandom functions[9]. The interest in this particular work stems not only from the novelty of the notion of indistinguishability, but because of the very natural construction based on an r-round Feistel network with independent random functions as round functions.

In a security proof involving PRFs the most difficult step is the analysis with an idealized primitive and proving that the resulting construction is information-theoretically indistinguishable from a another idealized primitive. Luby-Rackoff replace the PRFs in the Feistel construction with truly random functions and prove that the construction is indistinguishable from a random permutation when allowed only a bounded number of queries. Luby and Rackoff prove that 3-round random Feistel schemes are secure against all adaptative chosen plaintext attacks when the number of queries is $m \ll 2^{n/2}$, while 4-rounds are secure against chosen ciphertext attacks for the same number of queries[10]. For the latter case we say that the 4-round Feistel construction satisfies the *strong*-PRP property. Naor and Reingold [NR99] made the Luby-Rackoff construction simpler by replacing the first and last rounds by pairwise independent permutations.

We will now show that the 3-rounds Feistel (denoted as Ψ_3 from now on) construction is a pseudorandom permutation and then show it doesn't satisfy the strong PRP property by giving a concrete attack:

[8] There is also the notion of computational indistiguishability which holds against PPT adversaries.

[9] Aside note: interestingly we do not know how to construct a one-way permutation from a one-way function, and it is also believed that this is not possible under standard techniques.

[10] The bound $m \ll 2^{n/2}$ is called the "birthday bound", i.e. it is about the square root of the optimal bound against an adversary with unbounded computing power.

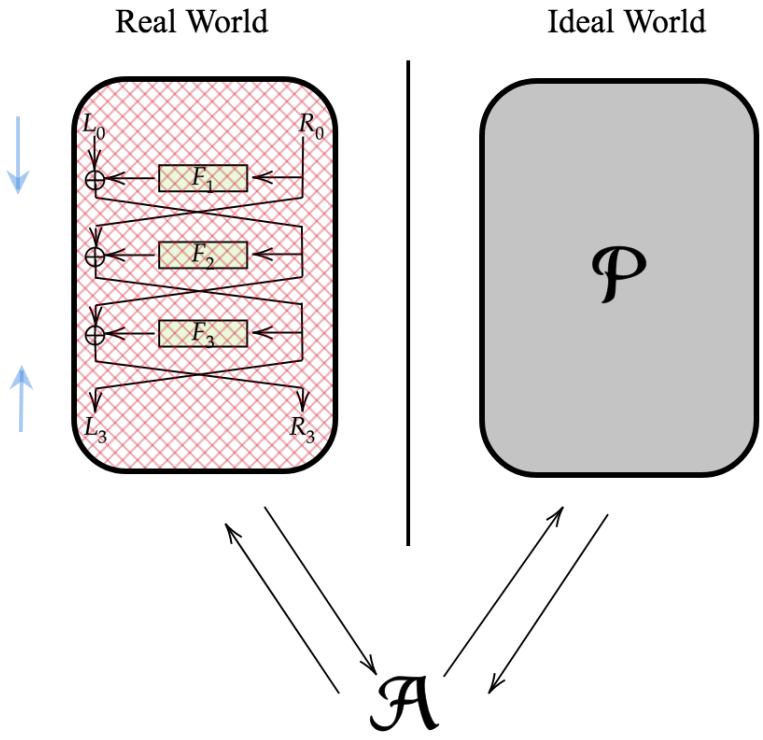

Fig. 3. Indistuinguishability of a Feistel construction. The distinguisher A that issues at most q queries, interacts either with the Feistel construction Ψ_3 (possibly backwards too) or a random permutation. The goal is to be able to tell with non negligible advantage, with which world he is interacting.

Proof. For our analysis, we model the round functions f_1, f_2, f_3 of the Feistel construction as PRFs on n-bits and we want to show that the advantage of any distinguisher that tries to tell apart Ψ_3 from a PRP π on 2n-bits issuing at most q queries to its oracle, is negligible. Formally we want:

$$\left|\Pr[D^{\Psi_3}(1^n) = 1] - \Pr[D^{\pi}(1^n) = 1]\right| \leq \text{negl}$$

Denote with (L_0^i, R_0^i) the i-th query that D asks the oracle and in the same fashion, denote by (L_3^i, R_3^i) the oracle's answer. Our proof strategy will be to show that for q queries, even when Ψ_3 is queried, the answers of the oracle will be uniformly distributed bits.

We define a collision at round k if we have $R_k^i = R_k^j$ for $i \neq j$ and we show that the probability of a collision in the first round is small:

If we have $R_0^i = R_0^j$ for some $i \neq j$ then in order for the entire inputs to be different, we have $L_0^i \neq L_0^j$. Then, at the end of the first round, the R_0 values will be XORed with not-equal L_0 values so the result will be something not-equal:

$$R_1^i = L_0^i \oplus f_1(R_0^i) \neq L_0^j \oplus f_1(R_0^j) = R_1^j$$

which means that because we think of f as a random function the probability of a collision after the first round is:

$$\Pr\left[L_0^i \oplus f_1(R_0^i) = L_0^j \oplus f_1(R_0^j)\right] = 2^{-n}$$

In q queries all distinct pairs of i, j are $O(q^2)$ so by union bound the probability of collision after the first round is $q^2/2^n$.

Now, conditioned on the fact that there was no collision in the first round, we prove that the probability of a collision in the second round is small. Following the previous reasoning and changing practically nothing, we know that since $R_1^i \neq R_1^j$ then $f_2(R_1^i)$ and $f_2(R_1^j)$ as outputs of random functions are independent and uniform which means that the probability of collision is:

$$\Pr\left[L_1^i \oplus f_2(R_1^i) = L_1^j \oplus f_2(R_1^j) \mid \text{no collision in the first round}\right] = 2^{-n}$$

Taking the union bound over distinct pairs (i, j) as in the first round, we get that the probability of not getting a collision on the second round given that there was no collision in the first round is $\leq q^2/2^n$.

Conditioning on the fact that there is no collision in the second round, we get that $L_3^i = R_2^i$ which are all independent and uniformly distributed and that the $R_3^i = L_2^i \oplus f_3(R_2^i)$ are also independent and uniform. Thus, we get back a $2n$-bit string in which the first n-bits are uniformly distributed and the second n-bits are uniformly distributed. When we query a pseudorandom permutation we get $2n$ uniformly distributed bits. The distinguisher's best strategy is to guess that it is interacting with a PRP when he sees $L_3^i = L_3^j$ for some $i \neq j$, something which happens with neglible probability. □

Additionally, a short intuitive proof is provided below, as to how the 3-round construction fails when the adversary is allowed access to a decryption oracle. Consider the following adversary A:

1. Query the decryption oracle for the decryptions of $0, 0$:
   $D(0||0) \to (x_1, x_2)$

2. Query the encryption oracle for $0, x_1$:
   $E(0||x_1) \to (y_1, y_2)$

3. Query the decryption oracle for $(x_2 \oplus y_2, y_1)$:
   $D(x_2 \oplus y_2, y_1) \to (x_3, x_4)$

4. If $x_3 = y_1 \oplus x_1$ output 1,
   else output 0

Claim. Adversary A can distinguish between a 3-round Feistel construction and a pseudorandom permutation with non-negligible probability.

Proof (of Claim). From the definition of the Feistel construction, we have the following values:

- $x_1 = F_{k_2}\left(F_{k_3}(0)\right)$
- $x_2 = F_{k_3}(0) \oplus F_{k_1}(x_1)$
- $y_1 = x_1 \oplus F_{k_2}\left(F_{k_1}(x_1)\right)$
- $y_2 = F_{k_1}(x_1) \oplus F_{k_3}(y_1)$

From the above it's easy to see that:

- $x_2 \oplus y_2 = F_{k_3}(0) \oplus F_{k_3}(y_1)$

So at the last carefully selected query we get:

$$\begin{aligned} x_3 &= y_1 \oplus F_{k_2}(x_2 \oplus y_2) \oplus F_{k_3}(y_1) = \\ &= y_1 \oplus F_{k_2}\left(F_{k_3}(0)\right) = \\ &= y_1 \oplus x_1 \end{aligned}$$

This is exactly as expected and is not a behavior that a pseudorandom permutation would exhibit. Thus, an adversary can easily distinguish between this construction and a pseudorandom permutation if allowed access to a decryption oracle. □

Patarin made improvements, showing that 7-round constructions achieve for CPA security when the number of queries is $m \ll 2^{n(1-\epsilon)}$ and 10 rounds are enough for CCA security [Pat03]. Additionally, Patarin studied random Feistel schemes when $m \ll 2^n$, showing that 5-round random Feistel schemes are CPA (see Section 2) secure and 6-round random Feistel schemes are CCA secure [Pat04]. Crucial to his methodology is the "H-Coefficients" technique, first introduced in [Pat91]. Below is a high level overview of this technique, as it is crucial for other proofs presented in this survey.

H-coefficients technique. As usual, consider an information-theoretic distinguisher, D, interacting with two worlds: the "real world" and the "ideal world". D maintains a list of all the queries made along with the responses returned, called a *transcript*. If $\mathcal{T}$ denotes the set of all possible transcripts, various elements of $\mathcal{T}$ have higher probability of appearing depending upon which world is responding to the distinguisher. The distinguisher then attempts to recognize which world he is interacting with on the basis of the recorded transcript. If X, Y is the distribution on the transcripts of the real world and the ideal world respectively, then D's advantage can be easily upper bounded by the statistical distance:

$$\Delta\left(X, Y\right) = \sum_{\tau \in \mathcal{T}} \left|\Pr[X = \tau] - \Pr[Y = \tau]\right|$$

The main observation of the technique is to bound $\Delta\left(X, Y\right)$ by using the fact that[11]:

$$\Delta\left(X, Y\right) = 1 - \mathbb{E}_{\tau \sim Y}\left[\min\left(1, \Pr[X = \tau]/\Pr[Y = \tau]\right)\right]$$

[11] Here, $\mathbb{E}_{\tau \sim Y}[Z(\tau)]$ denotes the expected value of the random variable $Z(\tau)$ when τ is sampled from Y and if $\Pr[Y = \tau] = 0$ then $\min\left(1, \Pr[X = \tau]/\Pr[Y = \tau]\right) = 1$

4.4 Indistinguishability of SPN constructions

We start this section by discussing a very important paper that gives candidate PRFs that are based on the SPN structure [MV15]. The importance of this work stems from the fact that it is the first attempt to define such PRFs, but also it is the first attempt to bridge the gap between the world of theory and the world of practice. As we have discussed, in theory we have security guarantees from some hardness assumption while in practice security is heuristic. In theory though the best PRF has key length quadratic to the length of the input [NR04], while in practice, the key length is about as big as the size of the input. In this work, the proposed PRFs are more efficient than previous constructions. In addition this is the first paper that gives an asymptotic analysis of the SPN structure.

Miles and Viola [MV15] take a complexity theoretic approach in defining security for SPNs. The authors give several candidate constructions for PRFs based on the SPN paradigm and more specifically the design of the AES S-boxes. The work analyzes linear SPNs where the underlying S-boxes where not necessarily permutations. In their construction the authors prove that r-rounds, for $r \geq 2$ are enough for CPA security. The bound worsens as the number of block cipher rounds increases. They also show security against linear/differential attacks for $r = \Theta(\log n)$ number of rounds. Below is an explanation of two of their most important candidate constructions.

The following notation is used and will make the description of the constructions easier:

- $b \in \mathbb{N}$, the S-box input size.
- $m \in \mathbb{N}$, the number of S-box invocations per round.
- $S : \mathrm{GF}(2^n) \to \mathrm{GF}(2^n)$, the S-box.
- $M : (\mathrm{GF}(2^n))^m \to (\mathrm{GF}(2^n))^m$, the linear transformation.

The first candidate F_1 is the first construction of a provably secure (but inefficient) PRF using the SPN design paradigm. It is an r-round SPN where the S-box is chosen uniformly at random from all the functions that map $\mathrm{GF}(2^n)$ to itself. The linear transformation M needs to have all non-zero entries (something which holds for any M with maximal branch number[12]). The resulting construction cannot be distinguished, by any adversary A, from a random function F:

Theorem 1. *For any adversary A that makes at most q queries to its oracle:*

$$\left|\Pr[A^F = 1] - \Pr[A^{F_1} = 1]\right| < O(r^2 m^3 q^3) \cdot 2^{-b}$$

[12] For a linear transformation $M : \mathbb{F}^m \to \mathbb{F}^m$ we define the branch number of M as

$$\mathrm{Br}(M) = \min_{\alpha \in \mathbb{F}^m \setminus 0^m} ((w(\alpha) + w(M(\alpha)))$$

where $w(\cdot)$ is the number of non-zero elements.

The techniques used are similar to [LR88] but since the S-box is not a permutation in this case, backwards queries are not allowed. At first it is counter intuitive, but the fact that the bound seems to get worse as the number of rounds increases but this is a feature of the specifics of their argument. By carefully examining the proof we see that only $r \geq 2$ is required.

The proof has two stages. In the same way that most of the proofs go, in the first stage it is shown that for a set of distinct queries $x_1, \ldots, x_q$, there exists a low probability bad event such that the output of the function is uniformly distributed if this bad event does not occur. By bad event we mean the case where two different queries induce the same output to some S-box in the final round. Formally:

Lemma 1. $\Pr[BAD] < O(r^2 m^3 q^3) \cdot 2^{-b}$

By bounding the probability of a bad event:

Lemma 2. *For any distinct $x_1, \ldots, x_q$ and any $y_1, \ldots, y_q$:*

$$\Pr\left[\forall i \leq q : F_1(x_i) = y_i | \neg BAD\right] = 2^{-qmb}$$

Now, conditioning on the fact that these bad events do not happen, the proof of the second lemma is fairly straightforward. The qm elements of the set $\{z_i^{(\ell)}\}_{i,\ell}$[13] are distinct and have not already been used. This means that every element has 2^{-b} probability of being mapped by S to a corresponding output.

The proof of the theorem follows immediately by those two lemmas when we consider the distribution of transcripts of A after its interaction with its oracles.

Example (A concrete instantiation). The only parameter we can fix is b since m will depend on it. As we discussed only $r \geq 2$ is required. So we can see that if we set $b = c \log n$ and $r = 2$ (and we restrict $q = poly(n)$ we get a PRF that is computable in time $n^{O(c)}$ and has security $n^{c'}$ for some $c' = \Omega(c)$.

The other interesting candidate of this work is F_4. We describe a modified version of F_4 briefly.

First we have that for the PRF candidate with seed $(k_0, k_1)^{2n}$, one round and a single S-box:

$$F_4(x) = (x + k_0)^{2^n - 2} + k_1$$

we can recover the seed with only four known input/output pairs. A short proof of this follows:

Proof (of Lemma 2). Let x_1, x_2, x_3, x_4 be inputs such that $x_1 + x_2 \neq x_3 + x_4$. Assume that $k_0 \neq x_i$ for all $i \in \{1, 2, 3, 4\}$ which happens with probability $(1 - 1/2^{n-2})$. If we write $y_i = F_4(x_i)$ then the equation:

$$(y_i + k_1) \cdot (x_i + k_0) = 1$$

[13] z_i denotes the state of the SPN's computation immediately before the final round of S-boxes for the i-th query.

is true for $i \in \{1,2,3,4\}$. Expanding the equation above we get:

$$k_0 k_1 + y_i k_0 + x_i k_1 + y_i x_i = 1 \tag{1}$$

If we sum equation 1 for $i = 1, 2$ we get:

$$(y_1 + y_2)k_0 + (x_1 + x_2)k_1 + (y_1 x_1 + y_2 x_2) = 0$$

because the quadratic terms cancel out.

By summing for $i = 3, 4$ we get another linear equation in k_0, k_1 so by solving them, we can extract the seed (k_0, k_1).

Even in the case where $y_1 + y_2 = y_3 + y_4$, we can recover k_1 because $x_1 + x_2 \neq x_3 + x_4$ and then recover k_0 from the initial equation. □

One very elegant aspect of this paper is the presentation of a candidate PRF which can be computed by a quasilinear circuit of size $\tilde{O}(n)$ while other theoretical constructions of PRFs have superlinear or even quadratic circuit size. This PRF is shown to have exponential security against a wide class of attacks, which makes it fall a bit short of the optimal (asymptotically) goal.

In later work, [BIP+18] gave the first candidate for an asymptotically optimal strong PRF, which means that super-linear circuit lower bounds cannot have natural proofs (a concept introduced in [RR97]). In particular that candidate is $F'_4 : \{0,1\}^n \to \{0,1\}$ which is:

$$F'_4 = \langle (x + k_0)^{2^n - 2}, k_1 \rangle$$

where the single S-box is combined with the Goldreich-Levin hardcore predicate [GL89]. This construction fools all parity tests that look at less than $2^{0.9n}$ outputs which is stated by the following theorem.

Theorem 2. *For any choice of $d \leq 2^n$, F'_4 is a d-wise small-bias generator with error $d/2^n$. For any distinct $a_1, \ldots, a_d \in \{0,1\}^n$ we have:*

$$\left| \Pr_{k_0,k_1} \left[\sum_{i=1}^{d} F'_4(a_i) = 0 \right] - \frac{1}{2} \right| < \frac{d}{2^n}$$

The theorem above is proven by showing that the polynomial $p(x) = \sum_{i \leq d} (a_i + x)^{2^n - 2}$ has at most $2d - 1$ distinct roots which means that when k_0 is not a root we have:

$$\Pr\left[\langle p(k_0), k_1 \rangle = 0\right] = \frac{1}{2}$$

which means that:

$$\left| \Pr_{k_0,k_1} \left[\sum_{i=1}^{d} F'_4(a_i) = 0 \right] - \frac{1}{2} \right| \leq \frac{1}{2} \Pr[p(k_0) = 0] < \frac{d}{2^n}$$

A fundamental shortcoming of the notion of indistinguishability is easily identifiable. A security proof using ideal primitives is only a heuristic indication of the security of the same scheme when using practical instantiation. For example, proving security in the random oracle model gives us certainty about the *design* of the scheme. As a potential break, the hash function used could not be well behaved enough. Moreover, there exist schemes that are considered secure in the random oracle model but completely break down when we replace the random oracle with any practical instantiation [CGH04].

To overcome this gap, a framework called *indifferentiability* [MRH04] was proposed. This framework is discussed in the following section.

5 Indifferentiability

As discussed in the previous section, the notion of indistinguishability is not enough to argue about the security of schemes and their proximity to idealized versions.

For this reason, a framework called *indifferentiability* [MRH04] was proposed. This framework provides a way to discuss the security of idealized constructions. For example, we can meaningfully talk about how "close" a block cipher construction is to an ideal cipher. More concretely, systems $\mathcal{S}$ and $\mathcal{T}$ are indifferentiable if the security of any cryptosystem using $\mathcal{T}$ as a component is not affected when $\mathcal{T}$ is substituted by $\mathcal{S}$. In addition, if $\mathcal{S}$ is differentiable from $\mathcal{T}$ then there exists a cryptosystem instantiated with one system that is secure, but becomes insecure when instantiated with the other system.

In order to fully understand this framework one can consider a block cipher based hash function that is *indifferentiable* from a random oracle in the *ideal cipher model*. Assume there is a scheme that utilizes a random oracle and is therefore secure in the *random oracle model*. Then, the scheme remains secure in the *ideal cipher model* after replacing the use of the random oracle with a hash function. We will treat this more formally in the upcoming section.

At a first glance, the notion of indifferentiablity seems to be very similar to the standard notion of indistinguishability that we saw in Definition 3. The drawback of indistinguishability is that it only applies in the black box case, where adversaries do not have any access to the inner workings of these systems. To resolve this, an extension to this definition called *indifferentiability*, where the adversary has access to public interfaces[14] of the system (a cryptosystem, or even a primitive), was developed. For example, a random oracle can be thought of as a *public* system. The interfaces to all participants are identical, thus the adversary has the same access as everyone else. If there is a system that an adversary does not have access to we say that that system has a *private* interface, accessible only to the honest parties.

> **Example.** To prove that a construction C -that uses a random function R- is "as good as" an ideal cipher E, it is necessary that C^R and E are indistinguishable, but it is not sufficient, as the adversary can exploit access to the oracle R.

[14] An interface is a way that the participants can interact with the system.

The previous example requires indistinguishability between $\left(C^R, R\right)$ and $\left(E, S^E\right)$ where S is a simulator with access to E. The simulator is a new concept explained here. The objective is to fool all distinguishers so that they cannot determine whether they are interacting with C^R, R or with the ideal cipher E. In the "world" of the ideal cipher there is no R, so a distinguisher would immediately know that he is not interacting with C^R, R. Therefore, for indistinguishability, the simulator is a system that will try to mimic the replies of R as closely as possible.

Definition 4. *A system C that has access to ideal primitives $\mathcal{P}$ is (t_S, q_S, ϵ)-indifferentiable from an ideal primitive $\mathcal{Z}$ if there exists a simulator S such that:*

$$\mathrm{Adv}^{\mathrm{indif}}_{C,\mathcal{Z},S}(D) = \left|\Pr\left[D^{C^{\mathcal{P}},\mathcal{P}} = 1\right] - \Pr\left[D^{\mathcal{Z},S^{\mathcal{Z}}} = 1\right]\right| \leq \epsilon$$

for all distinguishers D that make at most q queries to their oracles.

The simulator S runs in total time t_S and makes at most q_S queries to $\mathcal{Z}$. We note that t_S, q_S, and ϵ are functions of q.

The job of the distinguisher is to try and answer with non-negligible advantage probability whether it's interacting with one of the two following worlds (cf. Figure 4):

- The *real world* in which the distinguisher has access to the construction $C^{\mathcal{P}}$ and the primitives $\mathcal{P}$.
- The *ideal world* in which the ideal primitive $\mathcal{Z}$ has taken the place of $C^{\mathcal{P}}$ and the primitives $\mathcal{P}$ are substituted by the simulator S.

To prove indifferentiability, S is built in such a way that it succeeds in making $C^{\mathcal{P}}$ "look like" $\mathcal{Z}$ by trying to provide the proper replies to the queries that D sets to the primitives $\mathcal{P}$. In order to have any chance in succeeding at this task, S can query $\mathcal{Z}$ itself. To make this construction non-trivial, S cannot see the queries that D submits to $\mathcal{Z}$.

5.1 Indifferentiability of Feistel Networks

Feistel networks saw important results within the indifferentiability framework too. One of the most important papers in this line of work is due to Coron et al. [CHK+16] which combines results and techniques from [CPS08] and [HKT11]. There are two main results.

First, the authors prove that 5 rounds of Feistel are not indifferentiable (thus we need at least 6 rounds). Second, they prove that 14 rounds of Feistel are indifferentiable from an ideal cipher, thereby proving the equivalence of the ideal cipher model and the random oracle model.

The first result is shown by constructing a distinguisher that can find four inputs $(x_0, x_1), (x'_0, x'_1), (x''_0, x''_1), (x'''_0, x'''_1)$ with corresponding outputs $(x_5, x_6), (x'_5, x'_6), (x''_5, x''_6), (x'''_5, x'''_6)$ such that:

$$x_1 \oplus x'_1 \oplus x''_1 \oplus x'''_1 = 0 \ , \quad x_5 \oplus x'_5 \oplus x''_5 \oplus x'''_5 = 0$$

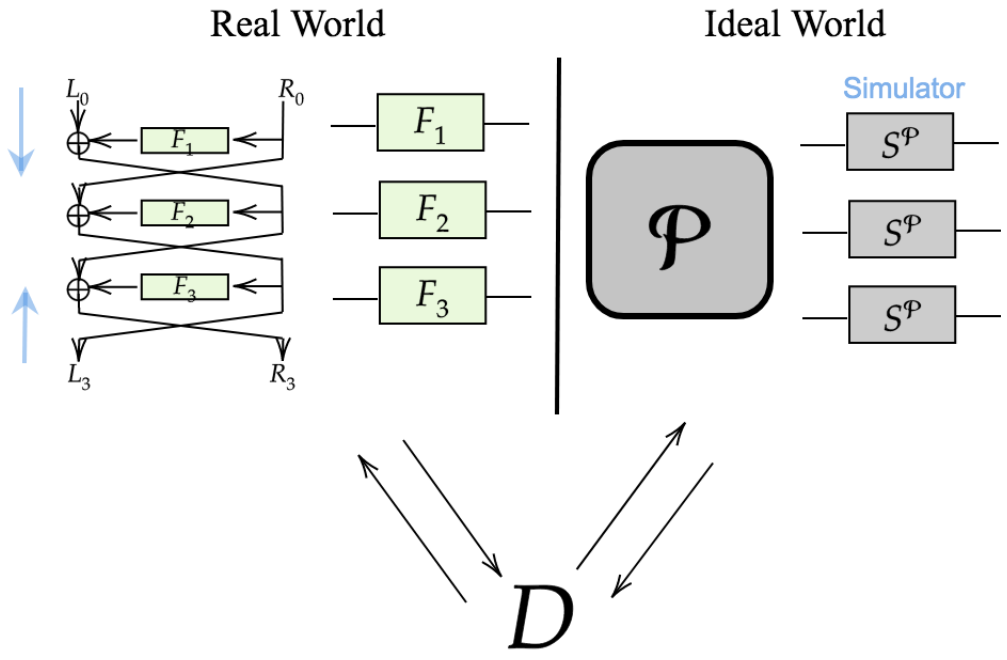

Fig. 4. Indifferentiability of a Feistel construction. The distinguisher D interacts either with the real world where he has access to Ψ_3 and the round functions F_1, F_2, F_3 or with the ideal world where he has access to a random permutation. In the ideal world there are no round functions so we use a simulator in their place. The simulator has access to the ideal primitive and tries to give answers to the distinguisher's queries that are consistent with the evaluations of the ideal primitive.

If the construction was indeed a random permutation then the task of finding such inputs would be hard and any efficient simulator would fail to create consistent answers to the distinguisher's queries. Thus, we have the following theorem that has a simple proof:

Theorem 3. *The 5-round Feistel construction using five independent random functions is not indifferentiable from a random permutation.*

Proof. We construct a distinguisher D as follows:

1. D chooses x_3, x'_3, x_4 arbitrarily (where x_i is the input to the round function F_i)
2. Computes $x_2 = x_4 \oplus F_3(x_3)$ and $x'_2 = x_4 \oplus F_3(x'_3)$
3. Compute:

$$\begin{cases} x_1 = x_3 \oplus F_2(x_2) \ , & x_0 = x_2 \oplus F_1(x_1) \\ x'_1 = x'_3 \oplus F_2(x'_2) \ , & x'_0 = x'_2 \oplus F_1(x'_1) \\ x''_1 = x'_3 \oplus F_2(x_2) \ , & x''_0 = x_2 \oplus F_1(x''_1) \\ x'''_1 = x_3 \oplus F_2(x'_2) \ , & x'''_0 = x'_2 \oplus F_1(x'''_1) \end{cases}$$

4. If x_1, x'_1, x''_1, x'''_1 are not pairwise independent, return 0.

5. Issue the following queries:

   - $(x_5, x_6) = P(x_0, x_1)$
   - $(x'_5, x'_6) = P(x'_0, x'_1)$
   - $(x''_5, x''_6) = P(x''_0, x''_1)$
   - $(x'''_5, x'''_6) = P(x'''_0, x'''_1)$

6. If $x_5 \oplus x'_5 \oplus x''_5 \oplus x'''_5 = 0$ return 1, else return 0.

When interacting with the ideal world, the probability that $x_5 \oplus x'_5 \oplus x''_5 \oplus x'''_5 = 0$ is less than $q^4/2^n$ so if q (the number of queries) is polynomial in n then this probability is negligible.

On the other hand, when interacting with the real world, since $x_3 \neq x'_3$ by definition then the probability that $F_3(x_3) \neq F_3(x'_3)$ is $1 - 1/2^n$. Conditioned on that we have that $x_2 \neq x'_2$ and by following the same logic for the rest we get that with probability $1 - 4/2^n$ the inputs x_1, x'_1, x''_1, x'''_1 are pairwise different. So when computing the Feistel construction we get:

$$\begin{cases} x_5 = x_3 \oplus F_4(x_4) \\ x'_5 = x'_3 \oplus F_4(x_4) \\ x''_5 = x'_3 \oplus F_4(x'_4) \\ x'''_5 = x_3 \oplus F_4(x'_4) \end{cases}$$

which means that always $x_5 \oplus x'_5 \oplus x''_5 \oplus x'''_5 = 0$ and we can always distinguish the real from the ideal world. □

The second important result can be summarized in the following theorem. As a corollary, that the random oracle model is equivalent to the ideal cipher model.

Theorem 4. *The 14-round keyed Feistel construction using a random oracle is indifferentiable from an ideal cipher. For an ideal cipher with κ-bit key and $2n$-bit inputs and any distinguisher making at most q queries, the simulator makes at most $1400q^8$ queries and runs in time $O(q^8)$*[15].

Although there was proof that a Feistel construction with 6 rounds and independent random round functions is indifferentiable from a random permutation in [CPS08], Holestein et al. in [HKT11] showed a distinguisher for the simulator used, effectively negating the result of Coron et al. On the bright side, based on its ideas and introducing their own ideas they managed to prove that a Feistel construction with 14 rounds is indifferentiable from a random permutation[16]. The proof is quite involved, spanning almost 20 pages and 30 lemmas. The proof can be summarized as follows:

[15] The distinguishing advantage is at most $\frac{10^8 q^{17}}{2^{2n}} + \frac{10^{22} q^{11}}{2^n}$

[16] The simulator used is very similar to the one used in [Seu09] which had an incorrect proof about the indifferentiability of 10-rounds Feistel construction.

We have 4 worlds that the distinguisher D interacts with: W_1, W_2, W_3, W_4. We start with the first world, W_1 where D interacts with (S, S^P) where S is the simulator and P is a random permutation. We aim to "reach" the final world W_4 where D will be interacting with a Feistel construction using random functions by making indistinguishable changes in between. As mentioned before, the simulator needs to enforce consistency of the values of $F_i(x_i)$ with P. A sequence of values $x_1, \ldots, x_r$ such that $F_i(x_i)$ is set by the simulator is called a chain.

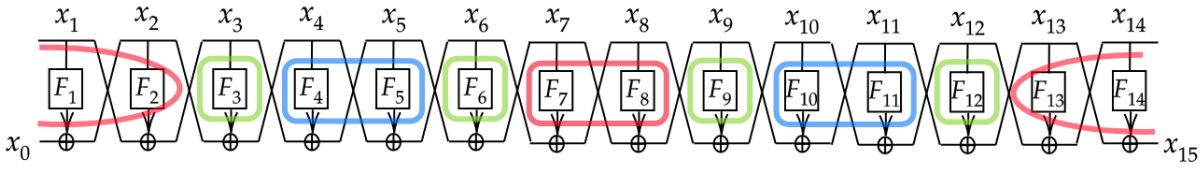

Fig. 5. The set uniform positions of the buffer zones are marked with green. The detect zones are marked with red, while the adapt zones are marked with blue. When the detect zone is filled, it triggers the completion of the entire chain.

To transition from W_1 to W_2 we replace P with a two-sided random function[17] R that uses randomness p possibly different than the simulator's randomness f. A two-sided random function is statistically indistinguishable from a random permutation, so R is indistinguishable from P.

To switch to W_3 we replace R with the 14-round Feistel construction which uses the same explicit randomness h as the simulator.

The simulation strategy (cf. Figure 5) considers only a carefully chosen set of partial chains[18]: In order for the simulator to preemptively find $P(x_0, x_1) = (x_{14}, x_{15})$, two *detect zones* are fixed. Detect zones, are sets of consecutive rounds $\{1, 2, 13, 14\}$, $\{7, 8\}$. Every time the simulator assigns a value to $F_i(x_i)$ it checks if there exists a tuple of the form $(x_1, x_2, x_{13}, x_{14}$ such that all $F_1(x_1), F_2(x_2), F_{13}(x_{13}), F_{14}(x_{14})$ have been assigned and $P\left(F_1(x_1) \oplus x_2, x_1\right) = (x_{14}, F_{13}(x_{13}) \oplus x_{14})$ or if a tuple of the form (x_7, x_8) exists such that $F_7(x_7)$ and $F_8(x_8)$ have been assigned. When there is a new query for F_i with input x_i, S sets $F_i(x_i)$ to a new random value and looks for the new relevant partial chains that involve x_i, adding them to a queue.

Then, until the queue is empty, S dequeues the first partial chain and completes it to a full chain $x_1, \ldots, x_r$ such that $P(x_0, x_1) = (x_{14}, x_{15})$[19]. When a partial chain includes both x_1 and x_14, we call it *wraparound*.

[17] A two-sided random function is very similar to a random invertible permutation. It stores two lists L_x and L_y and an invertible mapping from L_x to L_y. When queried on an element that it hasn't seen before (not in the list), it returns a uniformly random answer from $\{0,1\}^n$. In case a collision happens, the previous element is removed from the list and the mapping is updated accordingly.

[18] Contiguous subsequence of a chain.

[19] Here $x_0 = F_1(x_1) \oplus x_2$ and $x_{15} = F_14(x_14) \oplus x_{13}$

Example. Say that there is a query x_{14} which is answered with $F_{14}(x_{14})$ and then this output is used to query the first function $F_1()$. These queries would be a wraparound.

Another important aspect is the *4-round buffer zone* which is placed in order to ensure that overwrites do not happen. The simulator has 2 4-round buffer zones that correspond to rounds $\{3,4,5,6\}$ and $\{9,10,11,12\}$. In those buffers, positions $\{3,6\}$ and $\{9,12\}$ are the *set uniform positions* and the rest are called *adapt positions*. It is proven that when a chain is about to be completed, the set uniform positions are always unassigned. This ensures that that only after $F_3(x_3)$ is assigned, is x_4 determined so the probability that $F_4(x_4)$ has already been assigned is negligible.

One of the main parts of this proof is to show that the worlds $W_2(f,p)$ and $W_3(h)$ can be distinguished with negligible probability for uniformly random f,p,h. Will give a high level overview of it in the next paragraph.

In order to present this proof a bit succinctly, we will work conditioned on the fact that bad events do not happen. Additionally, we do not present the proof that randomly chosen (f,p) are good with high probability. We need the following claims:

1. No values in the table G_l are overwritten by the simulator[20].

2. After an execution of $W_2(f,p)$, for any primitive table entry $P(x_0,x_1)=(x_{14},x_{15})$, if we emulate the evaluation of the Feistel construction using tables G, we will also get (x_{14},x_{15}).

So now, conditioned on that executions in (f,p) are good we can provide a map between (f,p) and h, $\tau(f,p)=h$ which is the following: for any i and x let $h(i,x)=G_i(x)$ if $x\in G_i$, and $h(i,x)=\perp$ otherwise.

The claims allow us to prove that all the (query,answer) pairs to f (or h) by the simulator and the ones to R are indistinguishable making $W_2(f,p)$ indistinguishable from $W_3(\tau(f,p))$.

When the simulator sets $G_i(x)=f(i,x)$ in $W_2(f,p)$, then $G_i(x)=f(i,x)$ in the end of the execution (and thus by definition of τ we have $h(i,x)=f(i,x)$). The answer to a query $P(x_0,x_1)$ is exactly the same as the one given by the Feistel construction at the end of W_2. But each query that calls the Feistel construction for an evaluation but agree with h since we know that values are not being overwritten. This means that the query to P is answered the same by the Feistel construction in $W_3(h)$.

So when a query is issued by either the simulator or the distinguisher for $P(x_0,x_1)$ (backwards queries $P^{-1}(x_{14},x_{15})$ are handled in the same way) it gets the same answer in $W_2(f,p)$ and $W_3(h)$ and this concludes the proof.

After this work, which was focused on the simplicity of the proof rather than optimizing the number of rounds, the natural question to ask is: how small can the numbers of rounds be in order to achieve indifferentiability of the Feistel construction and the ideal cipher? Two closely published results gave positive answers. Firstly, in [DKT16], indifferentiability was

[20] $G_k(x)$ is the value stored for query x on round function F_k.

shown for 10-rounds. Then, in [DS16], indifferentiability was shown for 8-rounds. These two papers follow [CHK+16][21] closely, the differences in approach are outlined below.

In [DS16] small modifications are made to optimize the simulator used in [DS15][22] which part of it can be seen in Figure 6.

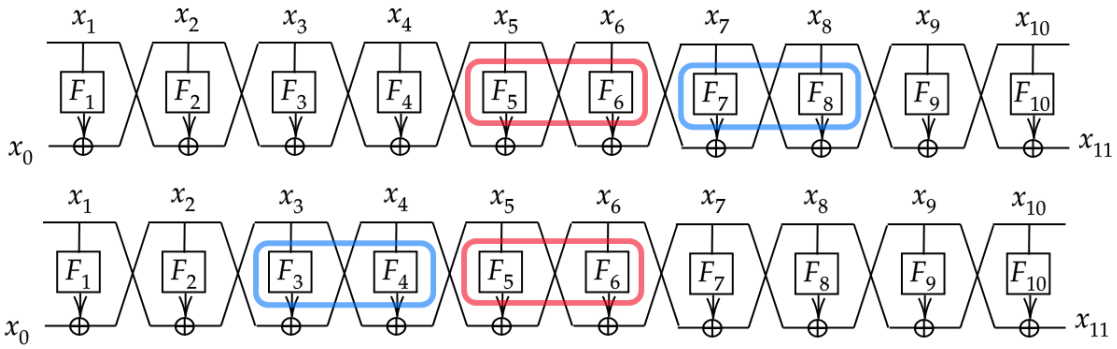

Fig. 6. The detect zone of the simulator is marked with red, while the adapt zone is marked with blue. When the detect zone is filled, it triggers the completion of the entire chain. When a query for F_6 fills the detect zone, then 7,8 becomes the adapt zone, while if a query for F_5 fills the detect zone, 3,4 becomes the adapt zone.

The main difference with the 14-round simulator is that they do not use separate buffer zones, thus easily reducing the number of rounds. The functionality of the buffer zone is still needed though so they use the rounds adjacent to the detect zones as their buffer zones. These are called the *endpoints*. We will provide a small example as to how the simulator works following the diagram in Figure 6.

Example. Lets say that the distinguisher asks for x_3. This value, either exists, and is returned, or it is being set randomly. Then the distinguisher asks for x_5 (which belongs to the detect zone) so this path is marked as pending but no completion is yet triggered. Say that the distinguisher asks for x_6. Now the detect zone is filled and the chain completion is being triggered. Because 7,8 are the adapt positions, the simulator will evaluate backwards the functions, at rounds 4,3,2,1 in that order. Then it will evaluate the entire permutation forward to get $x_1 1$ and will continue to evaluate positions 10, 9 and finally adapt positions 8,7 according to the permutation evaluation. If x_6 was asked first and then the distinguisher issued x_5, thus filling the detect zone from the right, the adapt zone would be 3,4 and the completion of the chain would be the opposite way than what was decribed before.

[21] Which in turn follows [Seu09]

[22] This concurrent work with Dachman-Soled et al. proved indifferentiability for 10 rounds but was not published.

The most impactful change to their 10-round simulator in order to reduce the number of rounds to 8 is that they split the middle detect zone into two bigger middle detect zones of three rounds each: $\{3, 4, 5\}$ and $\{4, 5, 6\}$. If the middle detect zones were kept the same (at rounds 4,5) then the queries that are adapated at those rounds would trigger new path completions of themselves.

There is small modification in the detection and completion of chains in Dachman-Soled et al. which makes the proofs a lot simpler. Their simulator operates in two phases, where in the first phase it only enqueues all partial chains which it thinks that will require completion and in the second phase it actually completes the chains and detects-enqueues only the middle detect zone. Secondly, in terms of the 4-round buffer zone, they allow the simulator to first complete the chains with the property that one of the set uniform positions has already been assigned.

We will now give an informal overview of the simulator. We denote by $F_1, \ldots, F_{10}$ the round functions by $F(i, x)$ we denote the query on x for function F_i. Whenever a query is issued, the simulator stores the query and the answer in the form of a pair (x, y) in 10 tables $G_1, \ldots, G_{10}$. When the simulator sees a query, $F(i, x)$ by the distinguisher it checks whether it's in table G_i and returns the stored value y. If it's a j-th new query, the simulator adds x to the set A_i^j.

Then the simulator checks whether i is one of the values $\{1, 2, 5, 6, 9, 10\}$ which are the endpoints of the detect zones and if that is the case, checks if there are new partial chains of the form $(x_9, x_{10}, 9)$, $(x_1, x_2, 1)$, and $(x_5, x_6, 5)$ that need to be enqueued. If it can't find any new partial chains then it picks a value uniformly at random and sets $G_i(x)$ to that value and returns it.

If it detects new partial chains that are enqueued in Q_{enq} then the simulator tries to evaluate them in a forward and backward sense as much as possible without setting any values in the table $G_{i'}$.

> **Example.** Say that the simulator stopped at $x_{i'} \notin G_{i'}$. The simulator will add $x_{i'}$ to $A_{i'}^j$ and will check if i belongs to any of the detect zone endpoints. From there, it will detect if $(x_{i'}, i')$ forms any additional partial chains and will enqueue them. This process is repeated until no partial chains are detected.

There are five queues that are being used to enqueue the chains for completion in that previous step: $Q_1, Q_5, Q_6, Q_{10}, Q_{all}$. The chains that are enqueued in Q_1, Q_5, Q_6, Q_{10} are those that have the *weak set uniform property* which is what we discussed before: allow the simulator to first complete the chains with the property that one of the set uniform positions has already been assigned. More concretely, if we have a chain $C = (x_k, x_{k+1}, k, l, g, b)$ that is enqueued to be adapted at position l, which means that the adapt positions in this case are l and $l + 1$ while the set uniform positions are $l - 1$ and $l + 2$ with the set uniform position that is adjacent to the query that caused C to be enqueued being at "good" set uniform position g and the other "set uniform" position at b. Note that b indexes the queues, Q_b. C is enqueued in Q_b if the value at the bad set uniform position b, x_b is not in G_b.

The first chains that are completed are the ones that are enqueued in the $\{Q_b\}$ queues. The simulator will evaluate the chain forward and backwards setting uniformly random values

at $G_i(x_i)$ that encounters and hasn't seen before. In the 4-round buffer that has the set uniform positions and the adapt positions, the simulator sets the values of the set uniform positions uniformly at random and forces the values at the adapt positions to be consistent with those of a random permutation.

When the simulator is done with the chains enqueued in $\{Q_b\}$ the simulator completes the chains in Q_{all}. The process is the same except that in this case the simulator detects partial chains of the form $(x_5, x_6, 5)$ and enqueues them in the queue Q_{mid}.

The last step is to complete all the chains in Q_{mid} and return the answer $G_i(x)$ to the query $F(i, x)$.

A natural observation that can be made is that indifferentiability results seem to proven for an even number of rounds. The answer for this is not definite and it probably has to do with the way the simulators are constructed. There is symmetry in the Feistel construction so the way the simulators of the papers described above work, we would have to adapt two queries each time. So starting with the initial proof of 14 rounds, each time there is a 2 round improvement.

5.2 Indifferentiability of SPNs

After having extensivelly surveyed the area of Feistel constructions, we switch our focus to the SPN paradigm. SPNs seem more important at the moment because of the immense usage of AES in practically every communication. Surprisingly, we have very few results on the security of SPNs.

Andreeva et al. [ABD+13], proved the indifferentiability of a 5-round key-alternating cipher from an ideal cipher in the random permutation model, where the key derivation function sets all rounds keys $k_i = f(K)$ where f is a random oracle. The importance of this design is that AES can be viewed as a 10-round key alternating cipher.

Let us define key-alternating ciphers before we present the main theorem:

Definition 5. *A key-alternating cipher, KA_t has t fixed permutations $P_1, ..., P_t$ on n bits, separated by key addition:*

$$KA_t(K, m) = k_t \oplus (P_t (...k_2 \oplus P_2 (k_1 \oplus P_1 (k_0 \oplus m))) ...)$$

where the keys $k_0, ..., k_t$ are called round keys and they are derived by the master key K according to some key schedule.

In the random permutation model, provable security results for this construction were first obtained for $t = 1$ round by Even and Mansour [EM91], who showed that the block cipher encrypting x into $k_1 \oplus P_1(k_0 \oplus x)$, where k_0 and k_1 are independent n-bit keys, is secure up to $O(2^{n/2})$ queries of the adversary. For this reason, this construction is often referred to as the Even-Mansour cipher.

In addition to the main result, Andreeva et al. also give attacks against KA_1, KA_2, KA_3. The attack for KA_1 is conceptually similar to the attacks on KA_2 and KA_3, but less involved. The claim is that $KA_1(K, x \oplus k_0) \oplus k_1 = KA_1(K', x \oplus k'_0) \oplus k'_1$ for any x, K, K' where $f(K) = (k_0, k_1)$ is an arbitrary key schedule.

Notice that in the real world, $KA_1(K, x \oplus k_0) \oplus k_1 = P_1(x \oplus k_0 \oplus k_0) \oplus k_1 \oplus k_1 = P(x)$ and similarly this is the case for K' so the equality holds with probability 1. In the ideal world, ideal cipher queries for $(K, x \oplus k_0)$ and $(K', x \oplus k'_0)$ return $u = \mathcal{IC}(x \oplus k_0) \oplus k_1$ and $v = \mathcal{IC}(x \oplus k'_0) \oplus k'_1$ the probability that $u = v$ is negligible for distinct keys K, K'. Thus, an adversary can distinguish between the two worlds.

The main result of this work is the following:

Theorem 5. *Let $P_1, \ldots, P_5$ be independent random n-bit permutations and f be a random function. Let D be an arbitrary, information-theoretic distinguisher that makes at most q queries. Then there exists a simulator S such that:*

$$\mathrm{Adv}^{\mathrm{indif}}_{KA_5,\mathcal{IC},S}(D) \in O\left(\frac{q^{10}}{2^n}\right)$$

where S makes at most $2q^2$ queries to the ideal cipher $\mathcal{IC}$ and runs in time $O(q^3)$.

As usual the proof starts from the real world and through a sequence of indinstinguishable changes, reaches the ideal world. A novelty of this work is that the probability of distinguishing between two worlds is not bounded with bad events. In places where a bad event flag might traditionally be used, the code simply aborts instead.

In Holestein et al. the map preserves exactly the probability of the execution and its image. In this work, the requirement is relaxed to *nearly equal probability*. This offers a more efficient and natural approach making the proof simpler. In addition, this work only considers the probability space that induced by the random footprints[23] instead of all possible random tapes. The rest, such as working with a distinguisher that completes all chains, is the same as the work of Holestein et al. The main lemma stated formally is:

Lemma 3. *For all n and all distinguishers D that interact either with W_2 or W_3, issuing at most q queries we have:*

$$\Pr[D^{W_2} = 1] - Pr[D^{W_3} = 1] \leq 160q^{10}/2^n + 81q^4/2^n$$

which is proven by the use of four other lemmas that span 12 pages.

To overcome the problem of the simulator-termination argument, i.e. to prove that the simulator has polynomial complexity, the authors utilize the concept of the "tripwire"[24].

A tripwire is an ordered pair $(i, i+1)$ or $(i+1, i)$ or $(1, 5)$ or $(5, 1)$ for a 5-round cipher. If a tripwire (i, j) is placed then the simulator will complete paths when it detects k-adjacencies[25] between positions i and j. If a tripwire is triggered, the simulator completes the relevant

[23] If $\alpha = (r_f, p_1, ..., p_5, p_E)$ is a W_2 random tuple, the footprint of α consists of that portion of the random tapes actually read during the execution $D^{W_2}(\alpha)$.

[24] The idea is due to [CPS08] but the term is new.

[25] We call k-adjacent a pair of queries $(1, x_1, y_1)$, $(5, x_5, y_5)$ if $k \in \mathcal{Z}$ and $E(f^{-1}(k), x_1 \oplus k) = y_5 \oplus k$. A sequence of queries $(1, x_1, y_1), (2, x_1, y_2), \ldots, (5, x_5, y_5)$ for which there exists a $k \in \mathcal{Z}$ such that each adjacent pair is k-adjacent and such that the first and last queries are also k-adjacent is called a completed k-path or completed k-chain.

chains and recursively completes chains for other potentially triggered tripwires; else it does nothing.

Dodis et al. [DSSL16] studied the indifferentiability of confusion-diffusion networks (which can be viewed as unkeyed SPNs). They proved that a constant round of confusion-diffusion rounds is sufficient to extend the domain of a public random permutation. The underlying permutations are modeled as both random and independent, a condition which seems justified by the fact that the proofs are already very involved. The simulator categorizes the confusion and diffusion rounds into 9 zones of four different types: one middle detect zone, left and right outer detect zones, four untangle zones, and two adapt zones.

Each one of them has one or more contiguous rounds and/or diffusion permutations, where every round and every diffusion permutation belongs to exactly one zone. There are three important takeaway points from this work. First, the security of the simulator is a function of the middle detect zone. Second, the query complexity is determined by the left outer detect zone and by the middle detect zone. Finally, the diffusion permutations must have low conductance.

Improving on [CS14], in which only asymptotically tight bounds were proven, in the indistiguishability setting, exact bounds on the security of key-alternating ciphers were given by Hoang and Tessaro [HT16], who showed that the r-round Even-Mansour construction is secure up to roughly $2^{rn/(r+1)}$ adversarial queries, when the public S-boxes are uniformly random and independent permutations and the round keys are independent.

We will provide the main combinatorial lemma from [CS14] because it is of independent importance and it is used in other works as a tool. This is a lengthy and complicated lemma so we need to first describe the setup.

Let G be a graph with $r+1$ shores[26] equal to $\{0,1\}^n$ indexed $0, 1, \ldots, r$. The edges of G are divided into r sets $E_1, \ldots, E_r$ where E_i is a (partial) matching between shores $i-1$ and i. We define U_{ij} for $0 \leq i < j \leq r$ to be the set of paths from shore i to shore j of G such that the vertex in shore i is left-free[27], but where the vertex in shore j may or may not be right-free.

For $0 \leq i \leq r$ we let u_i be a vertex chosen uniformly at random from the set of left-free vertices in shore i. The choice of $u_1, \ldots, u_r$ defines a path $w_0, w_1, \ldots, w_r$ in the following way: we set $w_0 = u$ and

$$w_i = \begin{cases} y & \text{if there exists an edge}(w_{i-1}, y) \in E_i \\ u_i & \text{otherwise} \end{cases}$$

We write $\Pr_G[u \to v] = Pr_G[w_r = v]$ for the probability that we arrive at vertex v in shore r by following this path.

[26] We associate to transcript τ a graph $G(\tau)$, which encodes the information contained in the key as well as in the permutations. $G(\tau)$ has $2(t+1)2^n$ vertices, grouped into "shores" of size 2^n each, with each shore being identified with a copy $\{0,1\}^n$

[27] A vertex in shore $i \geq 1$ is left-free if it is not adjacent to a vertex in shore $i-1$.

Lemma 4. *Let G and U_{ij} as described above. Then,*

$$\Pr_G[u \to v] = \frac{1}{N} - \frac{1}{N}\sum_{\sigma}(-1)^{|\sigma|}\prod_{j=1}^{|\sigma|}\frac{|U_{i_j i_{j-1}}|}{N - |E_{i_j}|}$$

The most significant contribution of [HT16] is not the result itself but an extension to the famous "H-coefficients" technique which started with [Pat91] and was heavily used by Chen and Steinberger in [CS14]. The extension is named the *expectation method* and a high level overview is provided in the following paragraph.

An important component for the *expectation method* is what the authors call *point-wise proximity*. That is, they show that for all possible transcripts τ, there exists an $\epsilon = \epsilon(q)$ such that for probabilities $p_0(\tau)$ (resp. $p_1(\tau)$) that the ideal world (resp. the real world) answer consistently with τ satisfy:

$$p_0(\tau) - p_1(\tau) \leq \epsilon \cdot p_0(\tau)$$

This means that that distinguishing advantage of any D is at most ϵ.

Point-wise proximity makes classical proofs techniques, like hybrid arguments and reductions, *transcript-centric*.

> **Example.** If we have worlds with probabilities p_0 and p_1 such that the previous inequality has been established, and we want to prove the same inequality for some other p'_0 p'_1 all we have to do is create a mapping ϕ such that:
>
> $$\frac{p'_1(\tau)}{p'_0(\tau)} = \frac{p_1(\phi(\tau))}{p_0(\phi(\tau))}$$

Although there is an exact formula for $\epsilon(\tau)$ in [CS14], ϵ depends on τ so in order to get a sharp bound, the authors increase the set of bad transcripts to include the ones that deviate a lot from their expectation and show a unique bound $\epsilon^* \geq \epsilon(\tau)$ for all good transcripts. In the absence of sharp concentration bounds, only Markov's inequality can be used so the bounds are not tight at all. To combine this with point-wise proximity Hoang and Tessaro use this ϵ that depends on the transcript for which the inequality holds and then, since the ideal world distribution is simple, replace the "better" ϵ^* with the expected value of $\epsilon(\tau)$.

More recently began of the study of SPNs as *strong* pseudorandom permutations where the underlying permutation is considered *public* [CDK+18]. When the permutation step is linear, as in the widely deployed AES, the authors give a general attack against any 2-round linear SPN when $w \geq 2$ and most importantly prove that 3-round linear SPNs are secure for all w if the underlying keyed permutations contain no zero entries and the first and last key are uniform.

Informally, the first and last rounds of a 3-round linear SPN can be thought of as blockwise universal permutations[28]. These proofs critically use the H-coefficients technique [Pat91]

[28] A keyed permutation π is blockwise universal if the following probabilities over uniform key k are small: (1) For any distinct x, x', $\pi(k, x) = \pi(k, x')$. (2) Two distinct blocks of $\pi(k, x)$ are equal. (3) $\pi(k, x) = c$ for a constant c

where they upper bound the probability of obtaining a bad transcript in the ideal world, which yields a final upper bound in the advantage of the distinguisher.

For the non-linear case, the following theorem is proved:[29]

Theorem 6. *Let $\delta, \delta^{'} > 0$, and let n and w be positive integers such that $w \geq 2$. Let T be a $(\delta, \delta^{'})$-super blockwise universal tweakable permutation. Then for any integers p and q such that $wp + 3w^2q < 2n/2$ one has:*

$$\mathrm{Adv}_{\mathrm{SP}^T}(p,q) \leq w^2q\left(\delta' p + \delta wq\right)\left(3\delta' p + 3\delta wq + 2\delta' wp\right) + \frac{q^2}{2^{wn}} + \frac{q\left(2wp + 6w^2q\right)^2}{2^{2n}}$$

which gives "beyond-birthday" security (for up to $2^{n/3}$ queries) for 2-round non-linear SPNs (with independent S-boxes and keys in different rounds). The proof relies on the modified version of the "H-coefficients" technique used in [HT16] and was described above.

6 Open Problems - Future work

There are some very interesting problems that we can identify after surveying this area. One obvious question would be whether we can prove indifferentiability of 6-rounds (or 7-rounds) Feistel constructions. An answer to this (for 6 rounds) would "match" the lower bound, since we know that 5 rounds are not enough, and would effectively close the chapter of indifferentiability of Feistel constructions.

Turning our focus to the less studied SPNs, one open question is whether r-round non-linear SPNs are enough to prove security up to $\mathcal{O}\left(2^{rn/(r+1)}\right)$. In addition, we would like to show beyond birthday security bounds for linear SPNs with $r \geq 3$ because they are the ones that are used in practice. Finally, on this direction, we want to prove tight security bounds and matching attacks for r-round linear and non-linear SPNs.

Inspired by the work of Miles and Viola, we should think about other choices of S-boxes apart from inversion because they might lead to more efficient constructions. By using other properties of linear transformation (other than the maximal-branch-number) we might be able to get stronger proofs of security. Combining these, could give a SPN that is computable by $O(n)$ size circuits.

Our ultimate goal would be able to construct even smaller primitives to build secure cryptographic cryptosystems. The motivation behind this idea can be seen since the early days and particularly the DES construction. DES, which is based on the Feistel construction, starts with r simple pseudorandom functions and by composing them, it achieves a specific level of security.

So if we start with a set of simple permutations[30] and we compose them for some number of times we seem to get a permutation that looks random. The question is: how many times do we need to compose those "simple" permutations in order to get a pseudorandom permutation? For some cases, the notion of pseudorandomness can be too much so we can relax it by asking how close the resulting permutation is to being k-wise independent.

[29] For easier parsing we note that SP^T is an r-round SPN based on a keyed tweakable permutation T

[30] By "simple" we mean that each output bit is only affected by a small part of the input.

Towards this research area we have the results of Hoory et al. [HMMR04] which give a bound on the number of times that simple permutations need to be composed in order for the resulting permutation to be "close" to being k-wise independent. Their result of $\tilde{O}\left(n^3k^3\right)$ compositions improves on the result of Gowers [Gow96] who had proved it for $\tilde{O}\left(n^3k(n^2+k)(n^3+k)\right)$ compositions [31].

The main idea in this line of work is to pick a very small number of bits that the permutation is going to change. To be precise, that number is 3 so these bits can define a cube consisting of eight elements. It is important to note here that the number of these permutations $O(n^3)$ is much smaller than the number of all possible permutations which is $(2n)!$. A permutation on this small cube is selected uniformly at random. It is then shown we do not need many compositions of these simple permutations in order to get a k-wise almost independent permutation. This means that for any $x_1, \ldots, x_k$ distint elements of a permutation π, the values $\pi(x_1), \ldots, \pi(x_k)$ are almost uniformly distributed.

Another conceptual contribution of [HMMR04] is the introduction of *strong closeness to k-wise independence* which in accordance to the works that we have surveyed, involves computationally *unbounded* adversaries. The adversary's task is to distinguish whether a permutation π that they are given oracle access to is a truly random permutation by issuing at most k queries.

Our hope is to build on this idea or even create different, more efficient primitives that can be used as building blocks for practical cryptosystems.

[31] Brodsky and Hoory later improved this bound [BH08].

Bibliography

[ABD+13] Elena Andreeva, Andrey Bogdanov, Yevgeniy Dodis, Bart Mennink, and John P. Steinberger. On the indifferentiability of key-alternating ciphers. In *Advances in Cryptology - CRYPTO 2013 - Proceedings, Part I*, pages 531–550, 2013.

[BH08] Alex Brodsky and Shlomo Hoory. Simple permutations mix even better. *Random Structures & Algorithms*, 32(3):274–289, 2008.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: - new simple PRF candidates and their applications. In *Theory of Cryptography Conference - TCC 2018 - Proceedings, Part II*, pages 699–729, 2018.

[Bir04] Alex Biryukov. The boomerang attack on 5 and 6-round reduced aes. In *International Conference on Advanced Encryption Standard*, pages 11–15. Springer, 2004.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Conference on Computer and Communications Security - CCS 1993*, pages 62–73, 1993.

[CDK+18] Benoît Cogliati, Yevgeniy Dodis, Jonathan Katz, Jooyoung Lee, John P. Steinberger, Aishwarya Thiruvengadam, and Zhe Zhang. Provable security of (tweakable) block ciphers based on substitution-permutation networks. In *Advances in Cryptology - CRYPTO 2018 - Proceedings, Part I*, pages 722–753, 2018.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.

[CHK+16] Jean-Sébastien Coron, Thomas Holenstein, Robin Künzler, Jacques Patarin, Yannick Seurin, and Stefano Tessaro. How to build an ideal cipher: The indifferentiability of the feistel construction. *J. Cryptology*, 29(1):61–114, 2016.

[CPS08] Jean-Sébastien Coron, Jacques Patarin, and Yannick Seurin. The random oracle model and the ideal cipher model are equivalent. In *Advances in Cryptology - CRYPTO 2008*, pages 1–20, 2008.

[CS14] Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology - EUROCRYPT 2014 - Proceedings*, pages 327–350, 2014.

[DKT16] Dana Dachman-Soled, Jonathan Katz, and Aishwarya Thiruvengadam. 10-round feistel is indifferentiable from an ideal cipher. In *Advances in Cryptology - EUROCRYPT 2016 - Proceedings, Part II*, pages 649–678, 2016.

[DS15] Yuanxi Dai and John P. Steinberger. Feistel networks: Indifferentiability at 10 rounds. *IACR Cryptology ePrint Archive*, 2015:874, 2015.

[DS16] Yuanxi Dai and John P. Steinberger. Indifferentiability of 8-round feistel networks. In *Advances in Cryptology - CRYPTO 2016 - Proceedings, Part I*, pages 95–120, 2016.

[DSSL16] Yevgeniy Dodis, Martijn Stam, John P. Steinberger, and Tianren Liu. Indifferentiability of confusion-diffusion networks. In *Advances in Cryptology - EUROCRYPT 2016 - Proceedings, Part II*, pages 679–704, 2016.

[EM91] Shimon Even and Yishay Mansour. A construction of a cioher from a single pseudorandom permutation. In *Advances in Cryptology - ASIACRYPT 1991*, pages 210–224, 1991.

[Fei73] Horst Feistel. Cryptography and computer privacy. *Scientific american*, 228(5):15–23, 1973.

[FKL+00] Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, and Doug Whiting. Improved cryptanalysis of rijndael. In *International Workshop on Fast Software Encryption - FSE 2000*, pages 213–230. Springer, 2000.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *Symposium on Foundations of Computer Science - FOCS 1984*, pages 464–479, 1984.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Symposium on Theory of Computing - STOC 1989*, pages 25–32, 1989.

[GM00] Henri Gilbert and Marine Minier. A collision attack on 7 rounds of rijndael. In *AES Candidate Conference*, volume 230, page 241, 2000.

[Gow96] Timothy Gowers. An almost m-wise independent random permutation of the cube. *Combinatorics, Probability and Computing*, 5(2):119–130, 1996.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HKT11] Thomas Holenstein, Robin Künzler, and Stefano Tessaro. The equivalence of the random oracle model and the ideal cipher model, revisited. In *Symposium on Theory of Computing - STOC 2011*, pages 89–98, 2011.

[HLL+00] Seokhie Hong, Sangjin Lee, Jongin Lim, Jaechul Sung, Donghyeon Cheon, and Inho Cho. Provable security against differential and linear cryptanalysis for the spn structure. In *International Workshop on Fast Software Encryption - FSE 2000*, pages 273–283. Springer, 2000.

[HMMR04] Shlomo Hoory, Avner Magen, Steven Myers, and Charles Rackoff. Simple permutations mix well. In *International Colloquium on Automata, Languages, and Programming - ICALP 2004*, pages 770–781. Springer, 2004.

[HT16] Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In *Advances in Cryptology - CRYPTO 2016 - Proceedings, Part I*, pages 3–32, 2016.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *Symposium on Foundations of Computer Science - FOCS 1989*, pages 230–235, 1989.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, 1988.

[Luc00] Stefan Lucks. Attacking seven rounds of rijndael under 192-bit and 256-bit keys. In *AES Candidate Conference*, pages 215–229, 2000.

[MRH04] Ueli M. 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 - Proceedings*, pages 21–39, 2004.

[MV15] Eric Miles and Emanuele Viola. Substitution-permutation networks, pseudorandom functions, and natural proofs. *J. ACM*, 62(6):46:1–46:29, 2015.

[Nat99] National Institute of Standards and Technology. *FIPS PUB 46-3: Data Encryption Standard (DES)*. October 1999.

[Nat01] National Institute of Standards and Technology. *FIPS PUB 197: Advanced Encryption Standard (AES)*. November 2001.

[NR99] Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-rackoff revisited. *J. Cryptology*, 12(1):29–66, 1999.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.

[Pat91] Jacques Patarin. Etude des g en erateurs de permutations bas es sur le sch ema du des. *Th ese de Doctorat de l'Universit e de Paris*, 6, 1991.

[Pat03] Jacques Patarin. Luby-rackoff: 7 rounds are enough for $2^{n(1-\epsilon)}$security. In *Advances in Cryptology - CRYPTO 2003 - Proceedings*, pages 513–529, 2003.

[Pat04] Jacques Patarin. Security of random feistel schemes with 5 or more rounds. In *Advances in Cryptology - CRYPTO 2004 - Proceedings*, pages 106–122, 2004.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

[Seu09] Yannick Seurin. Primitives et protocoles cryptographiquesa sécurité prouvée. *PhD thesis, Universite de Versailles Saint-Quentin-en-Yveline*, 2009.

[Sha49] Claude E Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.