Small-Box Cryptography

Yevgeniy Dodis ✉
New York University, USA

Harish Karthikeyan ✉
New York University, USA

Daniel Wichs ✉
Northeastern University, US
NTT Research, USA

Abstract

One of the ultimate goals of symmetric-key cryptography is to find a rigorous theoretical framework for building block ciphers from small components, such as cryptographic S-boxes, and then argue why iterating such small components for sufficiently many rounds would yield a secure construction. Unfortunately, a fundamental obstacle towards reaching this goal comes from the fact that traditional security proofs cannot get security beyond 2^{-n}, where n is the size of the corresponding component.

As a result, prior provably secure approaches — which we call *"big-box cryptography"* — always made n larger than the security parameter, which led to several problems: (a) the design was too coarse to really explain practical constructions, as (arguably) the most interesting design choices happening when instantiating such "big-boxes" were completely abstracted out; (b) the theoretically predicted number of rounds for the security of this approach was always dramatically smaller than in reality, where the "big-box" building block could not be made as ideal as required by the proof. For example, Even-Mansour (and, more generally, key-alternating) ciphers completely ignored the *substitution-permutation network* (SPN) paradigm which is at the heart of most real-world implementations of such ciphers.

In this work, we introduce a novel paradigm for justifying the security of existing block ciphers, which we call *small-box cryptography.* Unlike the "big-box" paradigm, it allows one to go much deeper inside the existing block cipher constructions, by only idealizing a small (and, hence, realistic!) building block of very small size n, such as an 8-to-32-bit S-box. It then introduces a clean and rigorous mixture of proofs and hardness conjectures which allow one to lift traditional, and seemingly meaningless, "at most 2^{-n}" security proofs for *reduced-round* idealized variants of the existing block ciphers, into meaningful, *full-round* security justifications of the actual ciphers used in the real world.

We then apply our framework to the analysis of SPN ciphers (e.g, generalizations of AES), getting quite reasonable and plausible *concrete* hardness estimates for the resulting ciphers. We also apply our framework to the design of stream ciphers. Here, however, we focus on the simplicity of the resulting construction, for which we managed to find a direct "big-box"-style security justification, under a well studied and widely believed eXact Linear Parity with Noise (XLPN) assumption.

Overall, we hope that our work will initiate many follow-up results in the area of small-box cryptography.

2012 ACM Subject Classification Theory of computation → Cryptographic primitives; Theory of computation → Problems, reductions and completeness; Theory of computation → Cryptographic protocols; Security and privacy → Information-theoretic techniques; Security and privacy → Mathematical foundations of cryptography; Security and privacy → Block and stream ciphers

Keywords and phrases Block Ciphers, S-Box, Cryptography

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.106

Funding *Yevgeniy Dodis*: Partially supported by gifts from VMware Labs and Google, and NSF grants 1619158, 1319051, 1314568.
Daniel Wichs: Partially supported by NSF grants CNS-1413964, CNS-1750795 and the Alfred P. Sloan Research Fellowship.

Acknowledgements The authors would also like to thank Stefano Tessaro for his help in applying the results of [31] in Section 4.4.

1 Introduction

Block ciphers are working horses of cryptography, and are used everywhere. Not surprisingly, we have many candidate constructions of block ciphers in the real world, including the industry-standard AES. The vast majority of such constructions iterate some relatively simple sequence of invertible transformations across multiple rounds and can be roughly divided into two main paradigms [28]: Feistel networks [21] or substitution-permutation networks (SPNs) [21, 35]. Simplifying somewhat, a Feistel round builds a keyed permutation on $2n$ bits from a "good" keyed round function on n bits; while an SPN round applies w "good" unkeyed permutations (so-called S-boxes) block-wise to its wn-bit input (for some $w \geq 1$), and then mixes the results with a keyed, non-cryptographic permutation on wn bits (called D-box). Examples of block ciphers based on Feistel networks include DES, FEAL, MISTY, and KASUMI; block ciphers based on SPNs include AES, Serpent, and PRESENT.

One of the biggest open problems in theoretical cryptography is to provide some theoretical justification about the security of this widespread approach of iterating "something simple" for many rounds. Ideally, such justification would be unconditional and provably secure. Unfortunately, obtaining such unconditional proofs is completely beyond our current capabilities (and would immediately imply $P \neq NP$, and more). The best we can do unconditionally (see [33] and the references therein) is to prove essential, but extremely limited, security properties of block ciphers, such as resistance to linear or differential attacks. While unconditional, these important results are insufficient for real-world applications of block ciphers to encryption and authentication. As the result, in order to prove sufficiently strong security properties of block ciphers, — such as security against chosen-plaintext/ciphertext attacks, — all existing approaches justifying security of current constructions roughly consist of 3 steps:

1. *Abstraction:* abstract and idealize some building block f inside the round function of the corresponding cipher.
2. *Proof:* show formal security of the resulting block cipher for some minimal number of rounds r, using a traditional reductionist approach.
3. *Conjecture:* make some kind of heuristic conjecture/assumption that, by increasing the number of rounds *well beyond* the minimal number of rounds r predicted in the prior step, existing real-world block ciphers are still secure, despite using much less idealized constructions of the building block f.

So far, existing realizations of this "recipe" used what we call a *big-box* approach to security. We detail this approach below in Section 1.1, where we show that it has several serious deficiencies in terms of our ultimate goal of building a block cipher from small components, such as cryptographic S-boxes. To address these problems, we introduce a novel paradigm for justifying the security of existing block ciphers, which we call *small-box cryptography*, described in Section 1.2. While the main motivation for small box-cryptography comes from the design of block ciphers, the framework is very general and can be used to build other primitives, such as hash functions, stream ciphers, pseudorandom functions, or even one-way functions. In particular, the framework consists of two main steps:

1. *Construction Step.* This step itself consists of two components specific to the primitive (e.g., block cipher, hash function, etc) we are building: *domain extension* and *hardness*

amplification. Despite being primitive-specific, it is largely *syntactic*, resulting in many constructions that have the potential to be secure in the real world.

2. *Analysis Step.* This step gives concrete exact security bounds/conjectures for the resulting constructions. It consists of three parts. The first two parts are *information-theoretic* and *fully provable.*[1] They formally analyze the domain extension and hardness amplification steps above within the existing techniques from "big-box" cryptography. The last step introduces a new "big-to-small" conjecture, which allows one to lift these big-box results to meaningful bounds/conjectures about the security of the resulting construction in the real world. In essence (see Theorem 14), this conjecture states if a natural-looking hardness amplification result gave a good security $\epsilon(n)$ against attackers running in time T assuming n is "large" ($n \gg \log T$, in particular), then the same construction will also have security $\epsilon'(n) \approx \epsilon(n)$ even for much smaller values of n, despite the fact that the supporting security proof critically breaks down in this case.[2]

We then apply our framework to the design of SPN-based block ciphers, which includes AES, Serpent, and PRESENT, among others. While the design of SPN ciphers is complex enough that we have no other ways to assess the soundness of our final security bounds, it appears that our bounds are (a) useful/practical; and, yet, (b) not contradicted by existing cryptanalysis. For example, instantiating our framework with a rather aggressive version of the "big-to-small" conjecture, we get can get the following concrete security bounds for generalization of AES (without key scheduler, for simplicity):

r-round variant of 128-bit AES with 8*-bit S-boxes is* $(2^{64}, (5.28)^{-r})$*-secure.*[3]

In particular, setting $r = 10$ (the number of real AES rounds), this would already yield respectable one-in-hundred-million security, while setting $r = 24$ would give excellent 2^{-64} security. Thus, to the best of our knowledge, our framework gives the most accurate and plausible theoretical justification for the security of SPN ciphers.

To complement our results, we also apply our framework to the design of pseudorandom generators (PRGs; aka stream ciphers). We then look at the resulting PRG construction, and analyze it *from scratch*, instead of applying the "Analysis Step" mentioned above (and, thus, avoid using the new and not-well-understood "big-to-small" conjecture; although we also analyze the resulting PRG in our new framework). Somewhat surprisingly, we show that not only did we get a meaningful PRG by blindly following the "syntactic" route, but the resulting construction was elegant enough to be analyzed using tools from big-box cryptography! In particular, we show that the resulting PRG is secure under the well-studied variant of the *Learning Parity with Noise* (LPN) assumption, called *Exact LPN* (XLPN) [27]. While the resulting "collision" of big- and small-box cryptography is likely a coincidence, it gives further evidence that the Construction Step of our framework often leads to plausibly-secure constructions, and motivates the further study of the "big-to-small" conjecture(s) introduced by this work.

---

[1] In practice, the hardness amplification step is often used with correlated round keys, using some "key schedule" heuristic. To model this case, we also need a plausible conjecture that the key schedule step does not violate the information-theoretic security proven using fully independent round keys.

[2] As we will see, the "big-to-small" conjecture looks very different from all previous ("big-box") hardness assumptions, and could be viewed as "one-way function" of small-box cryptography. While the particular conjectures introduced here might be too strong/aggressive or require further fine-tuning, the framework is general enough to accommodate future milder variants of this conjecture, still leading to meaningful real-world guarantees, while addressing the limitations of big-box cryptography.

[3] Here (T, ϵ)-security means that no T-time distinguisher can break the system with advantage greater than ϵ.

1.1 Big-Box Cryptography and Its Limitations

This approach follows the "abstraction-proof-conjecture" paradigm outlined above, but where the idealized building blocks f "big", meaning that its length n is proportional to block cipher length N. For example, the seminal paper by Luby and Rackoff [30] showed that a 4-round Feistel network yield a secure pseudorandom permutation on $N = 2n$ bits when applied to (independently keyed) round functions modeled as n-bit pseudorandom functions. Similarly, one can oversimplify the design of SPN ciphers, by ignoring its fine-grained substitution-permutation structure (arguably the "heart and soul" of the SPN design which goes back to Shannon [35]), — and instead view them as *key-alternating ciphers* [20, 5, 9, 25], where one models the entire SPN layer as a monolithic public permutation Π on $N = n$ bits. With such a higher-level abstraction, one can formally show that the r-round key-alternating cipher is secure, for any $r \geq 1$, in the *random permutation model* on N bits [20, 5, 9, 25], where $r = 1$ corresponds to the famous Even-Mansour cipher [20]. The advantage of the big-box approach is that one can formally prove exact security bounds which are exponentially small in the block length $N = O(n)$ of the underlying cipher E, and reduce the security of E to a slightly simpler building block f. Also, such proofs rule out certain generic attacks against the construction, and could generally be used as good "sanity checks" for the corresponding designs. However, they come with two significant disadvantages:

- First, since f is still "big', they do not come close to theoretically explaining how to build a block cipher from scratch, or, at least, from small components — which is the *ultimate goal* of block cipher design. In fact, one could subjectively argue that, in the existing constructions, the design of such a "large" component f is where "all the real action" is happening. For example, designing the round function of Feistel ciphers is, *by far*, the most intricate/interesting part of the design of DES, FEAL, MISTY, and KASUMI, where a wrong choice can render the whole design insecure. Similarly, completely ignoring the substitution-permutation structure of SPN ciphers (where the substitution is done by a small S-box, and permutation is a simple non-cryptographic D-box), once again ignores the heart of every SPN cipher, including AES.
- Second, the *actual* building blocks used by the existing constructions are *extremely* far from satisfying the idealized properties required for the provable security of this approach. For example, the round functions of DES and other Feistel ciphers are *nowhere close* to pseudorandom, while the simple 1-round SPN structure inside SPN ciphers is certainly *not* a random public permutation. As a result, it is completely unclear to what extent the provable results actually apply to the existing constructions. In fact, the number of rounds r sufficient for security with an idealized building blocks f is always dramatically lower than the number of rounds used (and needed!) in practice: there are no 4-round Feistel ciphers, or 1-round SPN (or key alternating) ciphers currently used.

To put it differently, while the "proof" part of the big-box approach can lead to good-looking bounds, the "abstraction" part is too coarse, while the "conjecture" part is really big (and also somewhat unclear). In particular, since none of the existing constructions have building blocks that are reasonably close to properties needed in theory, this approach does not give any guidance or explanation about why the particular real-world choices of implementing the "big-box" would be preferable to others, even with a *significantly increased* number of rounds. For example, the analysis of key-alternating ciphers does not shed any light as to why the round permutation build by the SPN structure is indeed much better than some affine permutation, which would be insecure, irrespective of the number of rounds. In other words, by keeping the box large, the big-box approach completely misses any theoretical explanation behind (arguably) the *most interesting* design decisions the practitioners must make when building actual ciphers.

1.2 Getting Closer to Reality: Small-Box Cryptography

To address the serious problems with the big-box approach outlined above, our new[4] approach attempts to go much deeper inside the existing block cipher constructions, by only idealizing a small (and, hence, realistic!) building block f, such as an S-box. For example, let us recall that an SPN cipher on wn bit inputs (where w is a relatively large constant $w \geq 1$), is computed via repeated invocation of two basic steps: a *substitution step* in which a public (unkeyed) "cryptographic" permutation $f : \{0,1\}^n \to \{0,1\}^n$, called an *$S$-box*, is computed in a blockwise fashion over the wn-bit intermediate state, and a *permutation step* in which a keyed but "non-cryptographic" permutation π on $\{0,1\}^{wn}$ is applied, called a *D-box*. Since π is non-cryptographic and typically linear, we will not idealize any of its properties, and work with D-box permutations π close to those used in practice. Hence, the only component which can be idealized is the S-box f, which we will model as a random permutation. Since the input length, n of f is small, such idealization is not unreasonable, which means the final construction analyzed is *really close* to what is used in practice, and certainly captures the heart of the SPN construction: namely, the *actual SPN structure*, as opposed to key-alternating ciphers, where this structure is completely ignored!

Of course, given the huge conceptual advantages of the small-box approach over the big-box approach in terms of the "abstraction" step, there is an important catch, as otherwise, we would likely have an unconditional result (and proved $P \neq NP$ along the way). The catch is that the best provable security one can conceivably get with such an approach is only exponential in n, as the S-box was the only idealized source of hardness that we could use. And since $n \ll N$ was very small by design (say, at most 32 in existing constructions), the actual bounds are not useful for practical use. At first, this admittedly serious deficiency appears to invalidate the whole point of provable security with this approach, which might have been the reason why so few papers followed this route prior to this work. However,

As one of the contributions of this work, we show that the seemingly useless bounds one gets in the "proof" component of the "small-box" approach, can still lead to very reasonable final results,

provided one properly models the "conjecture" component of this approach.

Small-Box Approach From the Sky. The approach is rather subtle and is carefully explained in Section 4. In brief, it formalizes two clean and explicit hardness conjectures, termed *hardness amplification* (Conjecture 13) and *big-to-small* (Conjecture 14). The hardness amplification conjecture, which is very plausible and can be sometimes proven even *unconditionally* (under appropriate independence assumptions) using a beautiful hardness amplification result of Maurer, Pietrzak, and Renner [31], states that the success probability ϵ of the distinguisher can be driven down exponentially by cascading the block cipher with itself.[5] Notice, such cascading is indeed a common practice of every block cipher design, where increasing the number of rounds (with independent or even correlated keys) is critical for

[4] As we detail in the related work Section 1.4, some of our ideas were already used in the prior work, but not in the totality that we present here.

[5] While we state this result for block ciphers, the framework of [31] is strong enough to study unconditional hardness amplification for other primitives, such as PRGs (where one XORs several PRGs with independent seeds).

improving the security of the block cipher. In particular, we can get this success probability to an extremely low level 2^{-wn} by cascading the original cipher $O(w)$ times.

However, this conjecture is only meaningful in the "big-box" setting, when the size n of our building block (e.g., S-box) is larger than the security parameter, as otherwise the exponential in n bounds given by our "proof component" are meaningless. To go back to the small-box case we care about, we notice that the success probability 2^{-wn} achieved in the big-box setting after cascading is also good and meaningful in the small-box case. In fact, the big-to-small conjecture states that even though the hardness amplification argument used to justify this conclusion crucially relied on the big-box assumption, the *final conclusion is actually true even in the small-box case*! Unlike the hardness amplification step, which appears very believable and even unconditionally true in certain settings, the big-to-small conjecture is completely new and not formally studied. However, despite being new and rather strong, it allows us to precisely state the kind of "leap of faith" one would be making when using constant size small-boxes.

We discuss these issues in more detail in Section 4, here only stating the end result of applying the 2 conjectures together. Here $n_0 = n_0(a, \alpha)$ is the constant defined in the big-to-small conjecture (and could be really small; $n_0 = 8$ in the case of AES), and we also don't explicitly state if cascading uses independent or correlated keys/building blocks (which is part of the hardness amplification conjecture):

► **Theorem 1** (Small-Box Cryptography; Informal). *Let T be the desired attacker time bound, and assume that r-rounds block cipher E of length wn utilizing idealized block f of size n is $(T, 2^{-\alpha n})$-secure, <u>as long as $n > a \log T$</u> (for some constants $a > 1$ and $\alpha < 1$). Then, under Conjectures 13 and 14, <u>for any $n \geq n_0(a, \alpha)$</u>, cascading E for $c = O(w/\alpha)$ times will result in a $r' = O(wr/\alpha)$-round block cipher E' which is $(T, O(T/2^{\ell(n)} + 2^{-wn}))$-secure,[6] where $\ell(n)$ is the key length of E' under to corresponding cascading step (equal to c times the key length of E when independent keys are used).*

The theorem above formalizes the last, "conjecture" step of small-box cryptography to get the following conclusion:

> *Under two clean and explicit hardness conjectures, one can get strong and meaningful security bounds for popular block ciphers, by obtaining "seeming useless" $(T, \mathsf{poly}(T)/2^n)$ security bounds for reduced-round variants of these ciphers with idealized building blocks of size n.*

Moreover, the small-box approach explicitly explains why the number of rounds r' used in practical constructions is *noticeably larger* than the theoretically predicted number of rounds r in the provably secure step: to drive the success probability of the distinguisher significantly below the minimum 2^{-n} level possible with the traditional information-theoretic proof. Thus, we have eliminated both significant disadvantages of the big-box approach: not guiding how to instantiate the "big" building blocks in practice, and giving inadequately low predictions for the number of rounds r needed for real-world security.

1.3 Our Results

We believe our main result is conceptual: bring the attention of the cryptographic to the deficiencies of "big-box" cryptography for the task of designing block ciphers and other

[6] For simplicity we consider uniform attackers; for other (e.g., non-uniform) models, we can change the conjectured $T/2^{\ell(n)}$ term to reflect the best generic attack in this model; see [12] for such non-uniform bounds for block ciphers.

symmetric key primitives, which are usually built from scratch, from very small components such as S-boxes. We also introduced a specific framework (which we called *small-box cryptography*) which is one concrete attempt to address this problem. This framework yields a rather syntactic way to derive candidate constructions conjectured to be secure in the real world and then proposes an explicit way to get concrete security bounds for the resulting constructions: by combining provably secure domain extension and hardness amplification steps with a new and unstudied type of hardness assumptions we call *"big-to-small" conjectures.*

We then apply this framework to the analysis of SPN ciphers (e.g, generalizations of AES), getting quite reasonable and plausible hardness estimates for the resulting ciphers. We also apply this framework to the design of stream ciphers. Here, however, we focus on the simplicity of the resulting construction, for which we managed to find a direct "big-box"-style security justification, under a well studied and widely believed XLPN assumption [27].

Overall, we certainly hope that our work will initiate many follow-up results in the area of small-box cryptography, which will both refine the initial heuristics (such as more refined analogs of our conjectured Theorem 1) outlined in this work, and add to a better understanding of existing symmetric-key constructions, hopefully well beyond block/stream ciphers.

1.4 Related Work

There are only a few prior papers looking at provable security of SPNs. The vast majority of such work analyzes the case of secret, key-dependent S-boxes (rather than public S-boxes as we consider here), and so we survey that work first.

SPNs with secret S-boxes. Naor and Reingold [34] prove security for what can be viewed as a non-linear, 1-round SPN. Their ideas were further developed, in the context of domain extension for block ciphers (see the further discussion below), by Chakraborty and Sarkar [8] and Halevi [24].

Iwata and Kurosawa [26] analyze SPNs in which the linear permutation step is based on the specific permutations used in the block cipher Serpent. They show an attack against 2-round SPNs of this form, and prove security for 3-round SPNs against non-adaptive adversaries. In addition to the fact that we consider public S-boxes, our linear SPN model considers generic linear permutations and we prove security against adaptive attackers.

Miles and Viola [33] study SPNs from a complexity-theoretic viewpoint. Two of their results are relevant here. First, they analyze the security of linear SPNs using S-boxes that are not necessarily injective (so the resulting keyed functions are not, in general, invertible). They show that r-round SPNs of this type (for $r \geq 2$) are secure against chosen-plaintext attacks.[7] They also analyze SPNs based on a concrete set of S-boxes, but in this case they only show security against linear/differential attacks (a form of chosen-plaintext attack), rather than all possible attacks, and only when the number of rounds is $r = \Theta(\log n)$.

SPNs with public S-boxes. The work of Cogliati *et al.* [11] analyzed SPNs with public S-boxes. In fact, this paper will basically give us the "domain extension" ($n \to wn$) component of our "Analysis Step", when we apply small-box cryptography to SPNs. Unlikely our work, however, the work of [11] did not advocate the hardness amplification to go beyond 2^{-n} security, or derived a concrete framework to assess the security of SPNs in the real world.

[7] In contrast, [11] showed that 2-round, linear SPNs are not secure against a combination of chosen-plaintext and chosen-ciphertext attacks when $w \geq 2$.

The earlier work by Dodis et al. [17] studied the *indifferentiability* [32] of confusion-diffusion networks, which can be viewed as *unkeyed* SPNs.

As observed earlier, the Even-Mansour construction [20] of a (keyed) pseudorandom permutation from a public random permutation can be viewed as a 1-round, linear SPN in the degenerate case where $w = 1$ (i.e., no domain extension) and all-round permutations are instantiated using simple key mixing. Security of the 1-round Even-Mansour construction against adaptive chosen-plaintext/ciphertext attacks, using independent keys for the initial and final key mixing, was shown in the original paper [20]. Kilian and Rogaway [29] and Dunkelman, Keller, and Shamir [18] showed that security holds even if the keys used are the same. As we mentioned, these results are insufficient for us, as we need a much larger (at least security parameter) domain expansion factor w.

Cryptanalysis of SPNs. Researchers have also explored cryptanalytic attacks on generic SPNs [2, 3, 4, 14]. These works generally consider a model of SPNs in which round permutations are secret, random (invertible) linear transformations, and S-boxes may be secret as well; this makes the attacks stronger but positive results weaker. In many cases the complexities of the attacks are exponential in n (though still faster than a brute-force search for the key), and hence do not rule out asymptotic security results. On the positive side, Biryukov et al. [2] show that 2-round SPNs (of the stronger form just mentioned) are secure against some specific types of attacks, but other attacks on such schemes have been identified [14].

Hardness Amplification. Harness amplification, going back to the seminal paper of Yao [37], amplifies the security of a given cryptographic primitive, typically by combining c independent copies of this primitives, and ensuring that the attacker must break all such copies. Traditionally, it is studied in the *computational setting* (e.g. [7, 6, 15, 10, 19, 23]), where one starts with (T, ϵ)-security, and gets (T', ϵ')-security, where $\epsilon' \approx \epsilon^c$. Unfortunately, such complexity-theoretic results, while extremely beautiful, have an inherent limitation that $T' \leq T\epsilon' \approx T\epsilon^c$. This means that the increased security comes at the price of a huge degradation in the run-time of the attacker, making these beautiful results completely useless for small-box cryptography. See [16] for more discussion.

Fortunately, hardness amplification has also been studied in the information-theoretic setting [31, 36], where the attacker is computationally unbounded but has a limited number of queries T to appropriate idealized oracles. In this setting, the security can be proven without much degradation in the parameter T, and this is the setting we use in our framework.

Random Local Functions. Goldreich [22] suggested designing a one-way function by repeatedly applying a certain local predicate f (which could be viewed as "S-box") to carefully chosen subsets of input bits. This influential work led to many follow-up constructions (see [1] and references therein) of how to build various "local" cryptographic primitives in this way, and argue about their security. At a high level, these results could be viewed as a different instantiation of small-box cryptography, which is incomparable to our proposal. Namely, our proposal focuses on capturing real-world designs where security is obtained by repetition and suggests modeling f as a random function/permutation in the Analysis Step. In contrast, the study of local cryptography is more focused on achieving small input locality (which is not our concern), as a result explicitly trying to avoid naive hardness amplification (which is expensive for locality). In other words, the two approaches happen to use "S-boxes" for completely different goals. It would be interesting to see if some interesting connection can be found between the two approaches to "small-box cryptography".

2 Applying Big-Box Cryptography to PRGs

In this section, we present our construction of a pseudorandom generator. We then prove its security under the eXact Linear Parity with Noise (XLPN) assumption. The construction, by itself, may not be the best PRG construction from this assumption, as it relies on large public parameters, which is unnecessary if one's goal to build a "big-box" PRG from XLPN. Of course, our point is to explicitly build and analyze cryptographic primitives from a "small" (but still polynomial size) S-box, which naturally mandates seemingly large parameters when viewed from the big-box perspective. Hence, the main purpose of our PRG construction is to introduce the small-box framework, before we look at the more complicated example of block ciphers in Section 4. In particular, unlike the case of block ciphers, the example will be simple enough that we can directly apply the "big-box" analysis to it (in the common reference string model, modeling our S-box).

2.1 Syntax and Security of PRG

A PRG is a primitive that is often used to produce random-looking string from a short, randomly chosen seed.

▶ **Definition 2** (Pseudorandom Generator). *Let $n \in \mathbb{N}$ be the security parameter. Then, an efficiently computable function $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ for $\ell(n) > n$ is an (T, ϵ)-secure PRG if for all adversaries $\mathcal{A}$ running in time T, the following holds:*

$$\left| \Pr_{s \leftarrow U_n} [\mathcal{A}(G(s)) = 1] - \Pr_{R \leftarrow U_{\ell(n)}} [\mathcal{A}(R) = 1] \right| \leq \epsilon$$

2.2 Our Construction

Recall, the goal of small-box cryptography is to analyze the direct construction of various primitives from "small" (constant- or polynomial-, but not exponential-) sized S-boxes. In the case of a PRG, it is natural to think of such an S-box as a Boolean function f modeled as a random function in the analysis. This is without loss of generality, as any non-Boolean S-box $f' : \{0,1\}^a \to \{0,1\}^b$ is equivalent to a Boolean S-box $f : \{0,1\}^{a+\log b} \to \{0,1\}$, where $f(x\|i)$ represents the i-th output bit of $f'(x)$. Further, it will be convenient for the notation to write the domain of this Boolean function as $\{0,1\}^{n+\log \ell}$, where ℓ is the desired output of our PRG, and n is the "small" leftover part. E.g., when $n = 8$ and $\ell = 256$, we get (still "small") 16-to-1 S-box.

For our "big-box" analysis, it will also be convenient to define a truth-table matrix for f as an $\ell \times N$ matrix $\mathbf{M}$, and think of this matrix as public parameters (or common random string, crs) of our PRG construction:

$$\mathbf{M} = \begin{pmatrix} f(1 \,\|\, 0) & \dots & f(N \,\|\, 0) \\ f(1 \,\|\, 1) & \dots & f(N \,\|\, 1) \\ \vdots & \ddots & \vdots \\ f(1 \,\|\, \ell - 1) & \dots & f(N \,\|\, \ell - 1) \end{pmatrix}$$

where $N = 2^n$.

Let $\mathcal{F} = \{f : \{0,1\}^{n+\log \ell} \to \{0,1\}\}$ be the set of all "S-box" functions f above. We now define a family of PRGs $\mathcal{G} = \{\tilde{G}_f : \{0,1\}^{nc} \to \{0,1\}^{\ell} \mid f \leftarrow \mathcal{F}\}$, which takes an additional "hardness" parameter c, and will expand a cn-bit input $x = (x_1, \ldots, x_c)$ into an ℓ-bit output y as follows:

$$y = \tilde{G}_f(x_1, \ldots, x_c) = \begin{pmatrix} f(x_1 \parallel 0) \oplus f(x_2 \parallel 0) \oplus \ldots \oplus f(x_c \parallel 0) \\ f(x_1 \parallel 1) \oplus f(x_2 \parallel 1) \oplus \ldots \oplus f(x_c \parallel 1) \\ \vdots \\ f(x_1 \parallel \ell-1) \oplus f(x_2 \parallel \ell-1) \oplus \ldots \oplus f(x_c \parallel \ell-1) \end{pmatrix}$$

Note on parameters. We need $\ell \geq nc + 1$ in order to ensure that our PRG is expanding, which lower bounds the domain length of the S-box by $(n + \log(nc + 1)) = O(\log c)$, if we think of $n = O(\log c)$. This is still a pretty good trade-off. Indeed, in both of our big- and small-box analyses (done in Sections 2.3 and 3), c will be the "security" parameter of the construction. So our security will scale — under appropriate hardness assumptions — exponentially in c. While the bit-size of the S-box input has only logarithmic dependence on the security parameter c. In particular, while the overall size of the S-box $\ell \cdot 2^n \approx c \cdot (n2^n)$ is noticeably greater than the PRG input size $c \cdot (n + \log \ell) \approx c \cdot (n + \log c)$, it is still polynomial in the security parameter c (assuming $n = O(\log c)$), and can be read by the attacker in its entirety.

2.3 Big-Box Analysis of $\tilde{G}$

In this section, we will undertake a big-box analysis of $\tilde{G}$ by proving its security from well-studied assumption, a variant of the LPN problem. The variant we consider is called the Exact LPN problem. This was first proposed and employed in proof of security by Jain *et al.* [27]. Much like the original LPN problem, the XLPN problem has a search and a decisional variant. It has been shown that the search variant of this problem is equivalent to the search version of the original LPN problem. Additionally, the hardness of the decisional XLPN problem is polynomially related to the search LPN problem.

▶ **Definition 3** (Decisional Exact LPN (XLPN) Assumption). *For $0 < \tau < \frac{1}{2}$, $q, m \in \mathbb{N}$, the (q, m)-XLPN_τ problem is (T, ϵ)-hard if for every adversary $\mathcal{A}$ running in time T, the following holds:*

$$\left| \Pr_{\mathbf{s},\mathbf{A},\mathbf{x}} \left[\mathcal{A}(\mathbf{A}, \mathbf{A}^\top \mathbf{s} \oplus \mathbf{x}) = 1\right] - \Pr_{\mathbf{A},\mathbf{y}} \left[\mathcal{A}(\mathbf{A}, \mathbf{y}) = 1\right] \right| \leq \epsilon$$

where $\mathbf{s} \leftarrow \mathbb{Z}_2^m$, $\mathbf{A} \leftarrow \mathbb{Z}_2^{m \times q}$, $\mathbf{x} \leftarrow \mathbb{Z}_{2,c}^q$ and $\mathbf{y} \leftarrow \mathbb{Z}_2^q$. Here, $\mathbb{Z}_{2,c}^q$ is the uniform distribution of q dimension binary vectors of weight $c = \tau \cdot q$.

To this end, we will prove the following theorem:

▶ **Theorem 4.** *Under the $(q = N, m = N - \ell)$-XLPN_τ assumption, the family of PRGs $\mathcal{G} = \{\tilde{G}_f : \{0,1\}^{nc} \to \{0,1\}^\ell | f \leftarrow \mathcal{F}\}$ is secure and provided $c = 2^n \cdot \tau$ and $\ell \geq nc + 1$, for $0 < \tau < \frac{1}{2}$.*

Discussion on parameters. Note that the length doubling PRG has an error-rate of $1/O(\log n)$, which is worse than a constant, but much better than $1/O(\sqrt{N})$ needed for public-key encryption. Finally, by suitably setting the parameters, we get the following result:

▶ **Corollary 5.** *For any polynomial N, let $\ell = N/2$ and $c = \ell/(2\log N) = N/(4\log N)$. Then, there exists a family of length-doubling PRG under the $(N, N/2)$-XLPN_τ assumption where $\tau = 1/O(\log N)$.*

We defer the proof of the above theorem to Section A. However, we discuss some instructive intuitions for the proof. Recall that in the PRG security game, the adversary $\mathcal{A}$ either receives $\tilde{G}(\mathbf{X})$ for $\mathbf{X} \leftarrow \{0,1\}^{nc}$ or $\mathbf{y} \leftarrow \{0,1\}^{\ell}$. To break this game, $\mathcal{A}$ would have to identify c values $x_1, \ldots, x_c$ that evaluates to the output that it has received, and in this setting $\mathbf{y}$ is a set of ℓ parity check equations.

In other words, if $\mathcal{A}$ finds a vector $\mathbf{x} \in \mathbb{Z}_2^N$ such that $wt(\mathbf{x}) = c$ and $\mathbf{Mx} = \mathbf{y}$, then with high probability, $\mathcal{A}$ received the real value and not the random value.

With this insight, it is useful to view this problem via the context of linear binary codes. In such a case, $\mathbf{M}$ can be considered as a parity check matrix and $\mathbf{y}$ is the syndrome of $\mathbf{x}$. However, this only works if $\mathbf{M}$ is of full row rank. Recall that a matrix $\mathbf{M}$ has a full row rank. if each of the rows of the matrix is linearly independent. Fortunately, we know that with overwhelming probability, a randomly sampled binary matrix has full rank.

In other words, given a random parity-check matrix $\mathbf{M}$ of size $\ell \times N$, we need to decode a random error vector $\mathbf{x}$, from the ℓ parity check equations, i.e., $\mathbf{Mx} = \mathbf{y}$, such that $wt(\mathbf{x}) = c$. Further, we get that $\binom{N}{c} < 2^{\ell} \implies c \log N < \ell < N$

Finally, given a parity-check matrix $\mathbf{M}$, one can efficiently calculate a corresponding generator matrix $\mathbf{A}$. Note that $\mathbf{A} \in \mathbb{Z}_2^{(N-\ell)\times N}$ and $\mathbf{MA}^{\top} = \mathbf{0}$, by definition.

3 Applying Small-Box Cryptography to PRGs

In the previous section, we presented the construction of a PRG, using an idealized primitive f, and proved its security under the XLPN assumption. In this section, we arrive at the same construction, but by religiously following the small-box framework. Recall, our recipe for small-box cryptography consists of two steps — the *construction step* and then the *analysis step*, each of which consists of several small steps. We detail each below.

3.1 Construction Step

The construction step of small-box cryptography consists of two smaller sub-steps: *domain extension* and *hardness amplification.* Although both of these steps are primitive-specific (e.g., different from PRGs and block ciphers), they are largely syntactic and require little-to-no technical expertise.

Domain Extension Step. Normally, the ideal object (S-box) gives a direct construction of the given primitive, but for "tiny" input/output domain. For example, in the PRG case the S-box $f : \{0,1\}^{n+\log \ell} \to \{0,1\}$ is a trivial "PRG" from $(n+\log \ell)$ bits to 1 bit. Of course, being non-expanding, this is not interesting in terms of functionality, but it will be obviously "secure" when we think of n as "big" and f as a "big" random oracle in subsequent sections.

To make the primitive interesting in terms of functionality even in the small-box world, the purpose of the domain extension step is to amplify the length of either the input, the output, or both to be large even in the "small" box world. In the case of PRG, the interesting parameter is the desired PRG output length ℓ, which we think as "big".[8] So our goal here is to extend the output domain from $\{0,1\}$ to $\{0,1\}^{\ell}$.

In the big-box world, one would amplify the output size by a factor of ℓ by expanding the PRG seed length by a factor of ℓ and concatenating the ℓ outputs of the base PRG. Here

[8] This explains our strange-looking choice of notation to denote the input length of our S-box as $(n+\log \ell)$ rather than just ℓ. Of course, this is just matter of convenience of notation: if the S-box size was n', we would have to subtract $\log \ell$ from it, and instead assume $n' = \log \ell + n$ for a new parameter n.

we do almost the same thing, except we don't need to pay in the seed length, and use our idealized modeling of our base PRG f as a random oracle rather than a "mere" PRG. This is consistent with the design intuition that a good S-box has all the idealized properties one would need for the construction to work. Namely, we can construct the range-extended PRG G as follows: $G : \{0,1\}^n \rightarrow \{0,1\}^\ell$:

$$G(x) = (f(x \parallel 0), \ldots, f(x \parallel \ell - 1)) \tag{1}$$

where $\parallel$ denotes concatenation. Intuitively, we simply "waste" $\log \ell$ bits of the seed to enumerate over the ℓ desired output bits.

Hardness Amplification Step. As we can see, the improved functionality — in this case, output size — came at the expense of decreased security (which is, of course, expected). For the PRG example above, the seed length was $(n + \log \ell)$ bits, but now is only n bits, which means it is definitely easier to break (we will formalize this quantitatively in Section 3.2).

The goal of the hardness amplification step is to amplify security — not just to the level we started from — but hopefully well beyond, so that we can afford to make n "small" and still have good looking security bound (this is somewhat subtle, and will be explained in the analysis step in Section 3.2). The hardness amplification step is usually parameterized by the hardness parameter c, which we can also think of as a security parameter of our final construction. For the case of PRGs, the standard hardness amplification is simply the bit-wise XOR operation, applied to c independent copies of our (already "domain-extended") PRG. Intuitively, while each individual PRG might only be slightly secure, by XOR-ing c independent copies the potential biases of the final PRG decay exponentially in c. This was formally analyzed in the computational setting by Dodis *et al.* [15] and in the information-theoretic setting by Maurer *et al.* [31].

With this in mind, we can define the following PRG $\tilde{G} : \{0,1\}^{nc} \rightarrow \{0,1\}^\ell$:

$$\tilde{G}(x_1, \ldots, x_c) = G(x_1) \oplus \ldots \oplus G(x_c)$$

This PRG can also be rewritten as follows, if we unwrap the definition of G from Equation (1):

$$\tilde{G}(x_1, \ldots, x_c) = \begin{pmatrix} f(x_1 \parallel 0) \oplus f(x_2 \parallel 0) \oplus \ldots \oplus f(x_c \parallel 0) \\ f(x_1 \parallel 1) \oplus f(x_2 \parallel 1) \oplus \ldots \oplus f(x_c \parallel 1) \\ \vdots \\ f(x_1 \parallel \ell - 1) \oplus f(x_2 \parallel \ell - 1) \oplus \ldots \oplus f(x_c \parallel \ell - 1) \end{pmatrix} \tag{2}$$

This is the same construction as the one in Section 2.2, but now obtained using two relatively syntactic steps. In each step, we intuitively think of f as a "big" random oracle to justify the soundness of this step (and we formalize this below), but the actual construction makes sense even in the "small-box" world! This dichotomy will be the point of the analysis step we present in the next section.

3.2 Analysis Step

On a high-level, the analysis step of small-box cryptography will consist of two components. The first component is *provable*, typically information-theoretically. It involves the analysis of the security of the final object ($\tilde{G}$, in the case of PRG, or SPN cipher in the case of block ciphers) in the corresponding idealized model for the building block f (random oracle model, in the case of PRG, and random permutation model in the case of SPNs). The proof will critically use the assumption that the size of f is larger than the running time T of the attacker $\mathcal{A}$ so that $\mathcal{A}$ cannot query f on all inputs. However, the final security bound one gets will be "syntactically meaningful" even in the small-box world, when the size of f becomes polynomial. Then the second component of the analysis will involve a new type of conjecture, which we term *Big-to-Small conjecture*, which was never considered prior to this work, and which allows one to get good exact security bounds for the final construction in the small-box world. We detail these below for the simple case of PRGs.

Idealized Big-box Proof. Here we are arguing the security of our final PRG $\tilde{G}$ in the random oracle model for the S-box f. Normally, one would try to do it modularly, by separately analyzing the domain extension step, followed by the hardness amplification step. Indeed, this is how we will do the analysis in the case of SPNs, where a direct analysis of the entire construction appears extremely cumbersome. Here, however, the PRG construction is so simple, that we do a direct proof for the security of $\tilde{G}$ in the random oracle model for f.

Recall that in the basic PRG security game, an adversary has to distinguish between $\tilde{G}(x)$ and a random ℓ-bit string, for a random seed $x = (x_1, \ldots, x_c)$, by making at most q queried to the random oracle f. We obtain the following simple lemma:

▶ **Lemma 6.** *Let $f : \{0,1\}^{n+\log \ell} \to \{0,1\}$ be modeled as a random oracle. Then, $\tilde{G} : \{0,1\}^{nc} \to \{0,1\}^{\ell}$ is $(q/N)^c$-secure PRG where $N = 2^n$, and q is the number of oracle queries made to f.*

Proof. Let us define the variable q_j to be the number of calls to f of the form $f(\cdot, j)$ for $j = 0, \ldots, \ell - 1$. Let $x_1, \ldots, x_c$ be n-bit strings, randomly sampled as the seeds. Now, define an event Bad_j as the event that a PPT attacker $\mathcal{A}$ invoked $f(x_1, j), \ldots, f(x_c, j)$. Now, note that the the probability that $\mathcal{A}$ invoked exactly one of these seeds with j is at most $q_j/2^n$. Therefore, $\Pr[Bad_j] \leq (q_j/2^n)^c$.

Define by $\mathcal{E}$ the event that any of $Bad_1, \ldots, Bad_{\ell-1}$ occurred. Then, we know that

$$\Pr[\mathcal{E}] = \sum_{j=0}^{\ell-1} \Pr[Bad_j] = \frac{1}{N^c} \sum_{j=0}^{\ell-1} q_j^c \leq \left(\frac{q}{N}\right)^c$$

Now, note that if $\mathcal{E}$ did not happen, then the adversary has no distinguishing advantage between real or random. Therefore, the distinguishing advantage of $\mathcal{A}$ in the PRG game is $(q/N)^c$. ◀

Removing the dependence on q in ϵ. We need one other syntactic, but extremely important step. For reasons to be clear when we move to the Big-to-small conjecture, we cannot afford to have a dependence on a number of oracle queries q in our security bound for ϵ. Instead, we will re-write our bound, but in a way that pushed the dependence on q into the lower bound for the S-box input parameter n. Concretely, if we (temporarily) assume that $n \geq 10 \log q$ (or, equivalently, $q \leq 2^{n/10}$), then $\epsilon(n) \leq 2^{-0.9nc} = N^{-0.9c}$.

Finally, we will now no longer assume that the attacker $\mathcal{A}$ is computationally unbounded, but instead upper bound its running time by some parameter $T \geq q$, and say that our PRG is (T, ϵ)-secure if no such attacker can break it with an advantage more than ϵ. With this change, we get the following restatement on our bound in Lemma 6 which will be convenient for our Big-to-small conjecture.

▶ **Theorem 7.** *If $n \geq 10 \log T$ and $f : \{0,1\}^{n+\log \ell} \to \{0,1\}$ is modeled as a random oracle, then $\tilde{G} : \{0,1\}^{nc} \to \{0,1\}^{\ell}$ given in Equation (2) is a $(T, N^{-0.9c})$-secure PRG, where $N = 2^n$.*

Big-to-Small Conjecture. Our analysis in the sections thus far have assumed that n is sufficiently large, i.e., "big n". Formally, Theorem 7 assumed that $n > 10 \log T$. However, the construction of $\tilde{G}$ is interesting even when n is much smaller. Indeed, we only need $cn < \ell$ to get a meaningful expansion. Moreover, even the final security bound $N^{-0.9c}$ is pretty good (while not established, of course!) for quire reasonable values of n and c. For example, setting $c = n = 8$ and $\ell = 128$, we get a PRG with seed length $cn = 64$, output length $\ell = 128$, and conjectured security $(2^{-64})^{-0.9} \approx 2^{-57}$, from a reasonably small Boolean S-box on 15 bits (or, equivalently, a more "balanced" S-box from 12-to-8 bits, which is quite reasonable to build). This would be fantastic, if true!

Of course, such security makes no sense, as it does not depend on the running time T of the distinguisher. Indeed, we could have replaced $n \geq 10 \log T$ with the bound $n \geq 1000000 \log T$, and basically get optimal security $\approx 2^{-nc}$ using a cn-bit seed, without doing any work. Nevertheless, we conjecture that bounds such as the one in Theorem 7 are hopefully meaningful for real-world security of the corresponding ciphers, provided one also includes some term corresponding to "brute-force attacks" running in time T. For example, the best generic (non-uniform) attacks against PRGs with cn-bit key [13] have an advantage roughly $T/N^{c/2}$ using non-uniform attackers using time and space T.

A particularly strong Big-to-small conjecture[9] would then state that the best way to attack constructions of the type we present is either by doing a brute-force search with advantage $T/N^{c/2}$ ignoring the fine-grained structure of our PRG, or we could have a generic attack on the structure of our PRG, ignoring its key size. And since with such a strong conjecture we have $T/N^{c/2} \gg N^{-0.9c}$, we are effectively saying that the brute-force attack is the best we can do for our cipher.

Of course, we could make weaker conjectures, and perhaps invest more time in the cryptanalysis of the resulting cipher. But the "mega-conjecture" of our approach is as follows:

> **Big-to-Small (Meta-)Conjecture**: *If the idealized big-box analysis shows $(T, N^{-\alpha c})$-hardness when $n > a \log T$ (for some $a > 1$ and $\alpha < 1$) for the c-time iterated construction of a given primitive, then the construction is also $(T, N^{-\alpha c} + \epsilon(T))$-secure for any $n \geq n_0$, where $n_0 = n_0(a, \alpha) \ll \log T$ is a constant, and $\epsilon(T)$ accounts for a term involving a brute-force search component in time T.*

► **Conjecture 8** (Big-to-Small Conjecture; Informal). *Assume a PRG G' of seed length $\ell(n)$ is $(T, \epsilon'(n))$-secure, where $\epsilon'(n) > T/2^{\ell(n)}$, when using ideal building component of length $n \geq a \log T$ (for some $a > 1$). Then, for some constant $n_0 = n_0(a)$, the "scaled down" version of G' of seed length $\ell(n_0)$ using building block f of size $n \geq n_0$ is still $(T, O(\epsilon'(n))$-secure.*

We defer a more precise discussion on such a conjecture, its practicality, and its impact after a similar analysis of SPNs in Section 4.3, as this is our most interesting case.

We note, however, that we would not be surprised that such a strong conjecture could be false in its generality. For example, analogous conjecture is clear false for related unpredictability primitives, such as one-way functions (OWF) constructed using direct product with independent inputs: $F(x_1, \ldots, x_w) = f(x_1), \ldots, f(x_w)$. Namely, when scaling the input length n to OWF f from security parameter to constant, we clearly make the resulting combined function F insecure, by iterative inverting each x_i one by one. However, it currently appears that funding natural counter-examples for indistinguishability primitives (like PRGs

[9] Of course, we have no chance of proving such a conjecture, as it clearly implies one-way functions.

and block ciphers) is quite non-obvious, even if one starts with artificial constructions not motivated by what is done in practice. Moreover, once the corresponding primitive is built using the natural hardness amplification step applied c times (e.g., cascade for block ciphers, or XOR for PRGs), the big-to-small conjecture becomes quite plausible. Indeed, we believe it could be true (while beyond our reach formally), at least with a weaker security term $N^{-a'c}$ for $a' < a$ (when the non-cascaded version has security N^{-a}). Further, the we would not be surprised if the brute-force component $\epsilon(T)$ could be improved by future cryptanalysis to be somewhat below the naive brute-force search.

To sum up, while many aspects of our framework are still being nailed down, we hope this work will motivate further explorations of small-box cryptography, including its promise and limitations.

4 Applying Small-Box Cryptography to SPNs

As our next result, we demonstrate the use of our framework to obtain concrete security bounds for SPN block ciphers.[10] In Section 4.1 we remind the reader of the syntax of (linear) SPNs. In Section 4.2 we show how we can obtain essentially the same construction by combining a "domain extension step" with the "hardness amplification" step. Namely, the former could be viewed as reduced-round SPN for which we will use the results of [11], which showed that 3-round linear SPNs achieve $O(T^2/2^n)$ security in the random permutation model (as a way to model the S-box, and under pretty mild restrictions on the linear D-box design). As stated before, a D-box is keyed, non-cryptographic permutation on wn bits. The latter step of "hardness amplification" could be viewed as cascading the cipher with independent (or correlated) keys to increase the number of rounds to get below 2^{-n} security barrier (in the "big-box" world). These analyses are done in Sections 4.3. Finally, Section 4.3 formalizes an appropriate "big-to-small" conjecture to go to the "small-box" world, and Section 4.4 brings everything together to justify Theorem 1 and get the concrete (conjectured) security bounds advertised in the Introduction.

4.1 Pseudorandom Permutations and SPNs

Pseudorandom Permutation. We now look at the security of a Pseudorandom Permutation (PRP).

▶ **Definition 9** (Pseudorandom Permutation). *Let $n \in \mathbb{N}$ be the security parameter. Then, an efficiently computable keyed-permutation $E_k : \{0,1\}^n \to \{0,1\}^n$ where $k \leftarrow \{0,1\}^s$ is an (T, ϵ)-secure PRP if for all adversaries $\mathcal{A}$ running in time T, the following holds:*

$$\left| \Pr_{k \leftarrow \{0,1\}^s} \left[\mathcal{A}^{E_k(\cdot)}() = 1 \right] - \Pr_{P \leftarrow \mathcal{P}} \left[\mathcal{A}^{P(\cdot)}() = 1 \right] \right| \leq \epsilon$$

where $\mathcal{P}$ is the set of all permutations over $\{0,1\}^n$. Note that if the construction uses an ideal object, then $\mathcal{A}$ gets oracle access to this primitive as well.

Substitution-Permutation Networks. A *substitution-permutation network (SPN)* is a keyed permutation defined by the two transformations that it repeatedly invokes. The first transformation is what is called an "S"-box where one computes, block by block, a public,

[10] Although we only apply our result to the SPN design, the discussion below is rather general, and can be applied to any r-round design E which uses some idealized building block f of (potentially small) size n.

cryptographic permutation. The second transformation uses a keyed, non-cryptographic permutation. The repeated invocation is determined by the rounds of the SPN. In addition, the distribution of the keys for the keyed-permutation is also included in this definition, though in practice, the keys are actually derived from a single master key through a *key schedule*.

Formally, an r-round SPN taking inputs of length wn where $w \in \mathbb{N}$ is the *width* of the network, is defined by:

1. $r+1$ keyed permutations $\{\pi_i : K_i \times \{0,1\}^{wn} \to \{0,1\}^{wn}\}_{i=0}^{r}$,
2. a distribution $\mathcal{K}$ over $K_0 \times \cdots \times K_r$, and
3. a permutation $f : \{0,1\}^n \to \{0,1\}^n$.

The actual construction is as follows:

- $x_1 := \pi_0(k_0, x)$.
- For $i = 1$ to r do:
  1. $y_i := \overline{S}(x_i)$, where $\overline{S}\,(x[1] \parallel \cdots \parallel x[w]) \stackrel{\text{def}}{=} f(x[1]) \parallel \cdots \parallel f(x[w])$.
  2. $x_{i+1} := \pi_i(k_i, y_i)$.
- The output is x_{r+1}.

where $(k_0, \ldots, k_r) \in K_0 \times \cdots \times K_r$ are the round keys and $x \in \{0,1\}^{wn}$ is the input.

Note that if f is efficiently invertible and each π_i is efficiently invertible (given the appropriate key), then one can simply reverse the process, given the round keys, to obtain the original input x.

Linear SPNs. In practice, majority of SPNs are what we call *linear*. Such SPNs correspond to the setting where the D-Boxes (i.e., the keyed permutations π_i) are defined as follows: $\pi_i(k_i, y) = k_i + y$, where each $k_i = T_i(k)$ with T_i being a linear transformation, and k being the "main" key. A simple example of such linear SPN corresponds to the case there each T_i is the identity function, meaning the original key $k = (k_0, \ldots, k_r)$ is $(r+1)wn$-bit long, and consists of $(r+1)$ independent sub-keys of length wn each. However, we could have more compact *key schedules* $T = (T_0, \ldots, T_r)$, where the main key k will be much smaller (and each function T_i possibly expanding). Indeed, such linear SPNs were analyzed by Cogliati *et al.* [11] (see Lemma 10 and Lemma 11 below).

Figure 1 is a pictorial representation of a 3-round Linear SPN with unspecified linear transformations T_0, T_1, T_2, T_3.

4.2 Construction Step

In this section, we show how the defined SPN can be "syntactically" obtained through a process of two steps — domain extension and hardness amplification.

Domain Extension Step. In this step, we view the S-box as an idealized block (random permutation), and our goal is to find the minimal number of rounds r for which SPNs (with appropriately chosen linear D-boxes) are $(T, 2^{-\Omega(n)})$-secure in the random permutation model. This is exactly the question studied by [11], who showed that minimal such $r = 3$, and we will use their concrete results in Section 4.3.

Hardness Amplification Step. First, since we are in the big world, we imagine the size n of the "small-box" f is made large enough so that exponential in n security is meaningful. For example, one could imagine SPN ciphers with large S-boxes (say, of several hundred bits long), even though they yield block ciphers of much higher block length wn than we

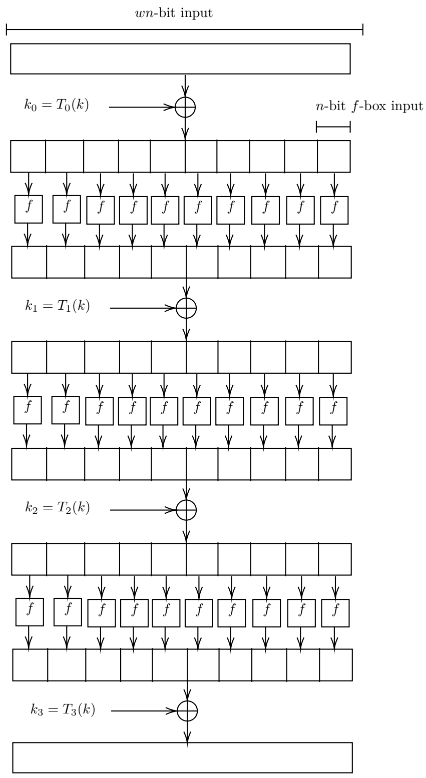

Figure 1 A 3-round Linear SPN with key schedule (T_0, T_1, T_2, T_3) expanding k to rounds keys (k_0, k_1, k_2, k_3), where $k_i = T_i(k)$ for $i = 0, 1, 2, 3$

might need (say, thousand bits or more). Then one can ask the question if the security of such "blown up" ciphers (still with idealized f) gets significantly better when one starts to increase the number of rounds r well beyond what is needed for their minimal security, by cascading the block cipher with itself, with independently generated keys. This is exactly the question of hardness amplification of block ciphers studied by [31, 36]; their result states that by cascading c independent, (T, ϵ)-secure ciphers, one still gets (T, ϵ')-security which decays exponentially in c: $\epsilon' \approx \epsilon^c$, but for our purposes any weaker exponential dependence on c (e.g., $\epsilon' = \epsilon^{c/100}$) will be enough to get a meaningful result, at the price of lesser efficiency. We give a more precise analysis in Section 4.3.

In summary, by doing this c-cascading step applied to the basic 3-round SPN predicted secure by [11] in the big-box world, we effectively obtain $3c$-round SPN, which was exactly our goal.

4.3 Analysis Step

Soundness of Domain Extension. As our next step, we analyze the soundness of hardness amplification in the big-box world, when we still model f as a "big" ideal object. As for the PRG case, we do it in the information-theoretic setting, where the running time of the attacker is unbounded, and only the number of oracle queries q is still bounded. Unlike the PRG case, the direct analysis of both domain extension and hardness amplification together appears extremely involved. Instead, we do it in a modular fashion, starting with the analysis of domain extension.

Fortunately for us, this question has been studied by Cogliati *et al.*[11]. They study the security of an SPN as a strong-pseudorandom permutation. Specifically, they show that a 2-round SPN is insecure with linear D-boxes but a 3-round SPN is secure, with caveats. Formally, these are the results for the 3-Round SPN which we present here, without proof. We invite the readers to refer to the original work for a complete discussion on the two Lemmas that we will use below.

▶ **Lemma 10** (Security of 3-Round SPN, Corollary 1 [11]). *For $w > 1$, there exists a 3-round linear SPN $k_0 = k_3 = k$ for uniform $k \in \{0,1\}^{wn}$ and set $k_1 = k_2 = 0^{wn}$ which is $\epsilon(q) = O(q^2/2^n)$-secure, where q is the number of queries made by the distinguisher.*

▶ **Lemma 11** (Security of 3-Round SPN, Corollary 2 [11]). *Let $w > 1$, k' be a uniform n-bit key, and a_i for $i = 1, \ldots, w$ are distinct non-zero elements of finite field $\mathbb{F} = \mathrm{GF}(2^n)$. Then, there exists a 3-round linear SPN with $k_0[i] = k_3[i] = a_i \cdot k'$, $k_1 = k_2 = 0^{wn}$ which is $\epsilon(q) = O(q^2/2^n)$-secure.*

Lemma 10 deals with the minimal security of the 3-round scheme. However, one can reduce the key length from wn to n (saving a factor of w), and Lemma 11 shows such reduction in key length still leaves the construction almost as secure, by utilizing a more aggressive key schedule.

Provable Hardness Amplification with Independent Keys. We begin by unconditionally *proving* the hardness amplification that we need (under appropriate independence assumptions) using a beautiful hardness amplification result of Maurer, Pietrzak, and Renner [31]. This is proved for a cascade of c block ciphers $E_1, \ldots, E_c$ which use both independent keys and independent ideal components f. For the case of SPNs, this means independent S-boxes with independent round keys. (We comment on how to relax this assumption later in the section.)

In the language of [31], imagine we have two indistinguishable "random systems" F and H, where:

- F provides two oracles, where the first oracle is the ideal building block f of length n, and the second oracle is the (keyed) block-cipher construction E_k^f utilizing f as an oracle and using a secret key k. Denote such block cipher by $E = E_k^f$, and $F = (f, E)$. Note, both forward and backward queries to E are allowed (and the same is true for f when f is a random permutation S-box).
- H provides two oracles, where the first oracle is still the ideal building block f of length n, but the second oracle is a random independent wn-bit permutation P. Denote such $H = (f, P)$. Note, both forward and backward queries to P are allowed (and the same is true for f when f is a random permutation S-box).

Assume further that no computationally unbounded distinguisher D making at most q queries to either F or H (for simplicity we do not split q into the number of primitive queries to f and construction queries to either E or P) can distinguish F from H with advantage greater than $\epsilon = \epsilon(q)$. Let us denote this by $\Delta_q(F, H) \leq \epsilon$.

Now, let $F_1, \ldots, F_c$ be c independent copies of F, and $H_1, \ldots, H_c$ be c independent copies of H. Let C be the construction such that, for $L_1, \ldots, L_c$ being each either F_i or H_i, $C(L_1, \ldots, L_c)$ implements $c + 1$ oracles, as follows. If we let $L_i = (f_i, Q_i)$ (where Q_i is either a random permutation P_i or E_i), then

$$C(L_1, \ldots, L_c) = (f_1, ..., f_c, Q_1 \circ Q_2 \circ ... \circ Q_c)$$

where $\circ$ is the composition of permutations. Namely, C is the c-time cascade of the c block ciphers E_i or random permutations P_i, which also provides oracle access to the c

independent building blocks $f_1, \ldots, f_c$. Let us also denote the c-cascade of our c block ciphers by $E' = E_1 \circ \cdots \circ E_c$, and the c-cascade of random permutations P_i by $P' = P_1 \circ \cdots \circ P_c$, which by itself is just another random permutation.

It is easy to see that this construction C has a property that is called *neutralizing* by [31]: whenever at least one of the H_i's is such that $L_i = H_i$ (the ideal system), meaning that Q_i is a fresh random permutation P_i, then

$$C(L_1, .., L_c) = (f_1, ..., f_c, P') = C(H_1, ..., H_c),$$

because the composition becomes random if at least one of the permutations is random. Under such conditions, the amplification result proven in [31] states that

$$\begin{aligned} \Delta_q(C(F_1, \ldots, F_c), C(H_1, ..., H_c)) &= \Delta_q((f_1, ..., f_c, E'), (f_1, ..., f_c, P')) \\ &\leq 2^{c-1}\epsilon^c < (2\epsilon)^c \end{aligned} \quad (3)$$

We can now apply Equation (3) to the 3-round linear SPN construction, where the building block f is an n-bit random permutation, and the security value $\epsilon(q) = O(q^2/2^n)$ is established by Lemma 10. We then get that the resulting $3c$-round SPN construction uses c independent S-boxes $f_1 \ldots f_c$ (one per each 3 rounds) and c independent wn-bit keys $K_1 \ldots K_c$, and achieves $(q, \epsilon'_c(q))$-security against q queries (to either the construction of the S-boxes), where $\epsilon'_c(q) = O((q^2/2^n)^c)$.

In fact, to reach the same conclusion with a shorter key length, we could use Lemma 11 in place of Lemma 10. In this case, we get the final key of length only $cn \ll cwn$, so we save the domain expansion factor w. Thus, although we still need c independent S-boxes, for now, this version and could be viewed as a relatively advanced form of key scheduling, with very strong provable security guarantees.

Removing the dependence on q in ϵ. As with the case of PRGs, we cannot use these results as is, and need to do some manipulation of the bounds to move the dependence on the number of queries q from ϵ on q to the size of the S-box f. Let $n \geq 20(\log q + 1)$ (or, equivalently, $2q^2 \leq 2^{n/10}$). Then $2\epsilon(n) = 2q^2/2^n = 2^{-0.9n}$, and hence $\epsilon'_c(q) \leq (2\epsilon(n))^c = 2^{-0.9nc} = N^{-0.9c}$.

Finally, we will now no longer assume that the attacker $\mathcal{A}$ is computationally unbounded, but instead upper bound its running time by some parameter $T \geq q$, and say that our SPN cipher is (T, ϵ)-secure if no such attacker can break it with an advantage more than ϵ. With this change, we get the following restatement on our bound above.

▶ **Theorem 12.** *If $n \geq 20(\log T + 1)$, then the $3c$-round SPN construction using c independent S-boxes and c independent (either wn-bit or n-bit, depending on variant) round keys is $(T, N^{-0.9c})$-secure.*

Conjectured Hardness Amplification with Correlated Keys. Unfortunately, the hardness amplification result of [31] crucially relies on the complete independence of the c S-boxes $f_1, \ldots, f_c$ and c independent round keys. In particular, unlike the much simpler PRG setting, where we managed to analyze the whole PRG construction in one go, for the case of SPNs, we currently cannot prove such strong results when the S-boxes are shared across the cascade, or keys are more correlated. The best provable result in this setting is the "computational hardness amplification" of Tessaro [36], but that comes with huge degradation in the number of oracle queries q allowed by the "cascade distinguisher", leading to concrete bounds which are not useful.

In general, though, we would like to apply an appropriate hardness amplification step in practical settings, where different cascading ciphers use correlated rather than independent keys (via a key schedule used in most actual designs), or when correlated or even identical building blocks f (e.g., S-boxes) are used in different cascaded ciphers. For such pragmatic settings, we do not have any provable results such as [31], and hence we state the hardness amplification step as a "conjecture" rather than "theorem" below. In particular, the concrete choice of cascading (not spelled out in the statement) is part of the conjecture. For simplicity, we also choose the final security level we desire to be 2^{-wn}, which is definitely enough for practical use, but the statement easily extends to any security level below 2^{-n}.

▶ **Conjecture 13** (Hardness Amplification; Informal). *Let T be the desired attacker time bound, and assume that r-rounds block cipher E of length wn utilizing idealized block f of size n is $(T, 2^{-\alpha n})$-secure,* as long as $n > a \log T$ *(for some constants $a > 1$ and $\alpha < 1$). Then,* provided $n > a \log T$, *cascading E for $c = O(w/\alpha)$ times will result in a $r' = O(wr/\alpha)$-round block cipher E' which is $(T, O(T/2^{\ell(n)} + 2^{-wn}))$-secure, where $\ell(n)$ is the key length of E' under to corresponding cascading step (equal to c times the key length of E when independent keys are used).*

Ignoring the cost of the brute-force key search (against uniform attackers, for simplicity) $T/2^{\ell(n)}$ (which is expected to be negligible for our choice of parameters), the hardness amplification conjecture states that using a building block f of size n would yield *better-than-exponential-in-n* security 2^{-wn} for sufficiently many more (still constant, assuming expansion $w = O(1)$) rounds, provided the box size n is sufficiently large.

Big-to-Small Conjecture. But now it seems natural to assume/conjecture that such a final result not only holds for "big" n but *might even be true for "small" n*! Namely, back to the original *small-box f*, we can reasonably conjecture security 2^{-wn} (plus brute-force search) for a sufficiently large constant number of rounds $r' = O(rw)$ *without assuming that this is only true when n is large.* Namely, the amplified security level 2^{-wn} is so good even if n is small, that we optimistically hope that it holds even in the small-box world, even though the supporting hardness amplification argument is no longer valid.

As discussed in Section 3.2, we will propose one of the strongest variants of such a conjecture. The motivation behind such a strong variant is that it gives us great security in case it happens to be true for practically used ciphers. As before, the conjecture will give a meaningful result for our purposes as long as one can decrease the size n of the 'small-box" below the threshold of $\log T$, for T independent of n. The constant $n_0 = n_0(a)$ below could be really small (e.g., $n_0 = 8$ in the case of AES), and is part of the conjecture. We also notice that we are *not* making this conjecture for all (even potentially artificial) block ciphers E', but only for specific E' resulting from applying the hardness amplification step to the basic block cipher E (for which we get our provably secure results).

▶ **Conjecture 14** (Big-to-Small Conjecture; Informal). *Assume a block cipher E' with key length $\ell(n)$ is $(T, \epsilon'(n))$-secure, where $\epsilon'(n) > T/2^{\ell(n)}$, when using ideal building component of length $n \geq a \log T$ (for some $a > 1$). Then, for some constant $n_0 = n_0(a)$, the "scaled down" version of E' using building block f of size $n \geq n_0$ is still $(T, O(\epsilon'(n))$-secure.*

We discuss this very strong conjecture below but notice that Conjectures 13 and 14 immediately imply the statement of Theorem 1 from the Introduction.

How Reasonable is "Big-to-Small" Conjecture? At first, this conjecture seems like a complete "cheat", as we simply assume that the conclusions attained by some security arguments crucially relying on the big-box assumption $n \gg \log T$, might still hold in the small-box world when n is a constant. But let us observe a couple of things. First, we already mentioned that we do not need such a strong conjecture: many weaker conjectures will yield meaningful variants of Theorem 1, provided they allow one to decrease the size n of the "small-box" below the threshold value $\log T$. Second, since the construction of E' is the same for all n, it is natural that its security smoothly changes with n, without any huge jumps at certain levels, as long as the exhaustive key search is infeasible (this is why we assumed $\epsilon'(n) > T/2^{\ell(n)}$). In particular, under this reasonable assumption, we certainly allow the assumed success probability $\epsilon'(n)$ to grow as the box f becomes smaller. So the only really big assumption is the fact that we kept the running time of the attacker at the same level T, even though when T becomes larger than 2^n, the attacker can suddenly evaluate our ideal component f (e.g., S-box) on *all* 2^n inputs. Third, given our current inability to built unconditionally block ciphers from only small components, it seems that some kind of "big-to-small" conjecture must be required, but we tried to make it as crisp and clean as we could, while additionally proving as many things around it as possible with the existing techniques. And, finally, the kinds of constructions we get when applying this conjecture to the SPN ciphers are exactly the SPN ciphers used in practice, and *believed to be secure*. So one can use this conjecture as a clean and formal way to isolate exactly the kind of "leap of faith" we are making in the real world in assuming these ciphers are secure.

Aside from these reasonable, but still rather limited, justifications at this stage we don't have any other theoretical justification for this strong "Big-to-Small Conjecture", and view this as an exciting direction for future research. In particular, given that coupling this strong conjecture with (rather mild and believable) hardness amplification step gives us the amazing conclusion of Theorem 1, which in turn implies plausible security for many SPN-based ciphers, we believe studying this new and non-standard conjecture is extremely reasonable and well-motivated.

4.4 Putting the Pieces Together

As mentioned earlier, Dodis *et al.*[11] proved results that addressed the problem of "domain extension" of block ciphers. In particular, they showed that a 3-round SPN is $(T, 2^{-\alpha n})$-secure when $n > 2\log T/(1-\alpha)$ (so that $T^2/2^n \leq 2^{-\alpha n}$). Thus, cascading it c times gives us $3c$-round SPN with conjectured $(T, T/2^{\ell(n)} + 2^{-\Omega(cn)})$-security, where $\ell(n)$ is our final key length, and this is true even for small values of n (governed by constant n_0 which is part of the conjecture). To get this close to the practical SPN designs, let us write $T = 2^t$, and assume we use correlated key schedule with final key length $\ell(n) = wn$, and, for simplicity, ideal hardness amplification is true even with best possible $\alpha \approx 1$. Then we get (very ambitious) conjectured $(2^t, 2^{t-wn} + 2^{-cn})$-security in $3c$ rounds. In particular, optimistically setting $n = 8$ and $wn = 128$ for the case of AES, we could get ambitious $(2^t, 2^{t-128} + 2^{-8c})$-security in $3c$ rounds. Assume $c \leq 8$ and $t = 64$ is good enough for practical use, we simplify this to an amazingly simple, but powerful, conclusion of our small-box cryptography framework:

$3c$-round variant of 128-bit AES with 8-bit S-boxes which is $(2^{64}, 2^{-8c})$-secure

In particular, setting $c = 10/3$, would already yield respectable one-in-hundred-million security in 10 rounds (the number of real AES rounds), while setting $c = 8$ would give excellent 2^{-64} security in 24 rounds.

While the above "back-of-the-envelope" calculations were a bit ad hoc and likely quite optimistic, they demonstrate several very attractive features of our framework, especially in comparison to its "big-box" counterpart. First, such calculations *can* be easily made (although more research is needed in estimating or conjecturing the right constants hidden/underspecified in Theorem 1). Second, such calculations give meaningful conjectured security of *actually used* ciphers. Third, for the first time, we see that our conjectured bounds

— even when ambitiously good — were on the *pessimistic* side, predicting either more rounds or a lower level of conjectured security than what is believed in practice. This is exactly what we expect from a sound theory, as we don't want such a theory to make predictions contradicted by reality.

5 Conclusion and Open Problems

We introduce the framework of *small-box cryptography*, which allows us to extend the (seemingly meaningless) provable security bounds for small values n into meaningful bounds for the iterated version of the corresponding cipher. Applying this framework to existing SPN ciphers, we get the most accurate theoretical justification for the security of these ciphers. While applying it to PRGs, we get a construction for which we can get an alternative proof from a well-studied assumption.

A number of interesting open questions remain. First, we have many open-ended questions regarding the soundness of our small-box approach, most important of which is a better understanding of the "big-to-small" Conjecture 14. It would also be interesting to apply the small-box framework to the Feistel ciphers, by going deeper into the design of its round function, so that we get much more meaningful justification regarding the design of existing such ciphers, including DES, FEAL, MISTY and KASUMI.

Second, it is interesting to understand the best way to get concrete security bounds using the current framework. For example, unlike the setting of "big-box" cryptography, where the improved security directly translates to smaller key length, in the setting of small-box cryptography the effect is much less understood, and likely significantly less important. For example, even proving optimal $O(q/2^n)$ security instead of $O(q^2/2^n)$ security for our reduced-round SPN simply changes the constant a from the hardness amplification Conjecture 13 from $a = 2/(1-\alpha)$ to $a = 1/(1-\alpha)$. This in turns might slightly decrease the minimal value of S-box size $n_0(a)$ in big-to-small Conjecture 14, but at the present we have no good understanding how practically important this change would be. In other words, proving "beyond-birthday" results in the small-box approach is certainly interesting on a technical level, but might not matter too much in terms of applying the framework to the existing ciphers.

References

1. Benny Applebaum. Cryptographic hardness of random local functions–survey. In Amit Sahai, editor, *Theory of Cryptography*, pages 599–599, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
2. Alex Biryukov, Charles Bouillaguet, and Dmitry Khovratovich. Cryptographic schemes based on the ASASA structure: Black-box, white-box, and public-key (extended abstract). In *Advances in Cryptology – ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 63–84. Springer, Heidelberg, Germany, 2014.
3. Alex Biryukov and Dmitry Khovratovich. Decomposition attack on SASASASAS. Available at http://eprint.iacr.org/2015/646.
4. Alex Biryukov and Adi Shamir. Structural cryptanalysis of SASAS. *Journal of Cryptology*, 23(4):505–518, 2010.
5. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, François-Xavier Standaert, John P. Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 45–62. Springer, Heidelberg, Germany, 2012.

6 Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In Joe Kilian, editor, *TCC 2005: 2nd Theory of Cryptography Conference*, volume 3378 of *Lecture Notes in Computer Science*, pages 17–33, Cambridge, MA, USA, February 10–12, 2005. Springer, Heidelberg, Germany.

7 Ran Canetti, Ronald L. Rivest, Madhu Sudan, Luca Trevisan, Salil P. Vadhan, and Hoeteck Wee. Amplifying collision resistance: A complexity-theoretic treatment. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 264–283, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Heidelberg, Germany.

8 Debrup Chakraborty and Palash Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. In *Fast Software Encryption – FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 293–309. Springer, Heidelberg, Germany, 2006.

9 Shan Chen and John P. Steinberger. Tight security bounds for key-alternating ciphers. In *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 327–350. Springer, Heidelberg, Germany, 2014. `doi:10.1007/978-3-642-55220-5_19`.

10 Kai-Min Chung, Feng-Hao Liu, Chi-Jen Lu, and Bo-Yin Yang. Efficient string-commitment from weak bit-commitment. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 268–282, Singapore, December 5–9, 2010. Springer, Heidelberg, Germany.

11 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*, volume 10991 of *Lecture Notes in Computer Science*, pages 722–753. Springer, 2018. `doi:10.1007/978-3-319-96884-1_24`.

12 Sandro Coretti, Yevgeniy Dodis, and Siyao Guo. Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 693–721. Springer, 2018. `doi:10.1007/978-3-319-96884-1\_23`.

13 Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and PRGs. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 649–665, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Heidelberg, Germany.

14 Itai Dinur, Orr Dunkelman, Thorsten Kranz, and Gregor Leander. Decomposing the ASASA block cipher construction. Available at http://eprint.iacr.org/2015/507.

15 Yevgeniy Dodis, Russell Impagliazzo, Ragesh Jaiswal, and Valentine Kabanets. Security amplification for interactive cryptographic primitives. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 128–145. Springer, Heidelberg, Germany, March 15–17, 2009.

16 Yevgeniy Dodis, Abhishek Jain, Tal Moran, and Daniel Wichs. Counterexamples to hardness amplification beyond negligible. In *Proceedings of the 9th International Conference on Theory of Cryptography*, TCC'12, page 476–493, Berlin, Heidelberg, 2012. Springer-Verlag. `doi:10.1007/978-3-642-28914-9_27`.

17 Yevgeniy Dodis, Martijn Stam, John P. Steinberger, and Tianren Liu. Indifferentiability of confusion-diffusion networks. In *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 679–704. Springer, Heidelberg, Germany, 2016.

18 Orr Dunkelman, Nathan Keller, and Adi Shamir. Minimalism in cryptography: The Even-Mansour scheme revisited. In *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 336–354. Springer, Heidelberg, Germany, 2012.

19 Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 342–360, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany.

20 Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. In *Advances in Cryptology – ASIACRYPT'91*, volume 739 of *Lecture Notes in Computer Science*, pages 210–224. Springer, Heidelberg, Germany, 1991.

21 Horst Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15–23, 1973.

22 Oded Goldreich. *Candidate One-Way Functions Based on Expander Graphs*, page 76–87. Springer-Verlag, Berlin, Heidelberg, 2011.

23 Oded Goldreich, Noam Nisan, and Avi Wigderson. On yao's xor-lemma. In *In Electronic Colloquium on Computational Complexity (ECCC*, 1995.

24 Shai Halevi. Invertible universal hashing and the TET encryption mode. In *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 412–429. Springer, Heidelberg, Germany, 2007.

25 Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 3–32. Springer, Heidelberg, Germany, 2016. doi:10.1007/978-3-662-53018-4_1.

26 Tetsu Iwata and Kaoru Kurosawa. On the pseudorandomness of the AES finalists—RC6 and Serpent. In *Fast Software Encryption – FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 231–243. Springer, Heidelberg, Germany, 2000.

27 Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *ASIACRYPT*, volume 7658, pages 663–680. Springer, 2012. URL: https://www.iacr.org/archive/asiacrypt2012/76580656/76580656.pdf, doi:10.1007/978-3-642-34961-4_40.

28 J. Katz and Y. Lindell. *Introduction to Modern Cryptography, 2nd edition.* Chapman & Hall/CRC Press, 2015.

29 Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive key search (an analysis of DESX). *Journal of Cryptology*, 14(1):17–35, 2001.

30 Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

31 Ueli M. Maurer, Krzysztof Pietrzak, and Renato Renner. Indistinguishability amplification. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 130–149, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Heidelberg, Germany.

32 Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, Heidelberg, Germany, 2004.

33 Eric Miles and Emanuele Viola. Substitution-permutation networks, pseudorandom functions, and natural proofs. *Journal of the ACM*, 62(6):46, 2015.

34 Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999.

35 Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28(4):656–715, 1949.

36 Stefano Tessaro. Security amplification for the cascade of arbitrarily weak PRPs: Tight bounds via the interactive hardcore lemma. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 37–54, Providence, RI, USA, March 28–30, 2011. Springer, Heidelberg, Germany.

37 Andrew C. Yao. Theory and applications of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91. IEEE Computer Society Press, 1982.

A Proof of Theorem 4

Proof. With the above intuition, we can prove the hardness amplification result through a sequence of hybrids, and reducing the problem to a variant of the LPN problem. In the proof we denote the uniform distribution of binary vectors of length N and weight c by $\mathbb{Z}_{2,c}^N$.

Hybrid H_0. $\mathcal{A}$ receives $\mathbf{Mx}$ for $\mathbf{x} \leftarrow \mathbb{Z}_{2,c}^N$ and $\mathbf{M} \leftarrow \mathbb{Z}_2^{\ell \times N}$.

Hybrid H_1. $\mathcal{A}$ receives $\mathbf{Mx} \oplus \mathbf{MA}^\top \mathbf{s}$ where $\mathbf{A}$ is the generator matrix corresponding to the parity check matrix $\mathbf{M} \leftarrow \mathbb{Z}_2^{\ell \times N}$. $\mathbf{A} \in \mathbb{Z}_2^{(N-\ell) \times N}$, $\mathbf{s} \leftarrow \mathbb{Z}_2^{N-\ell}$, and $\mathbf{x} \leftarrow \mathbb{Z}_2^N$ with $wt(\mathbf{x}) = c$

Note that Hybrids H_0 and H_1 are identically distributed because of the property that $\mathbf{MA}^\top = \mathbf{0}$

Hybrid H_2. $\mathcal{A}$ receives $\mathbf{Mx} \oplus \mathbf{MA}^\top \mathbf{s}$ where $\mathbf{M}$ is the parity check matrix corresponding to the generator matrix $\mathbf{A} \leftarrow \mathbb{Z}_2^{(N-\ell) \times N}$ $\mathbf{s} \leftarrow \mathbb{Z}_2^{N-\ell}$, and $\mathbf{x} \leftarrow \mathbb{Z}_2^N$ with $wt(\mathbf{x}) = c$

Note that the difference between Hybrids H_1 and H_2 only lies in the order of sampling $\mathbf{M}, \mathbf{A}$. In H_1, we sample $\mathbf{M}$ and then compute $\mathbf{A}$, while in H_2 we do the opposite.

Hybrid H_3. $\mathcal{A}$ receives $\mathbf{Me}$ where $\mathbf{M}$ is the parity check matrix corresponding to the generator matrix $\mathbf{A} \leftarrow \mathbb{Z}_2^{(N-\ell) \times N}$ and $\mathbf{e} \leftarrow \mathbb{Z}_2^N$.

▷ Claim 15. If $(N, m = N - \ell)$-XLPN$_\tau$ is (t, ϵ)-hard, then the distinguishing advantage between H_2 and H_3 for any PPT adversary $\mathcal{A}$ is at most ϵ provided $c = N \cdot \tau$

Proof. Let us assume that there is $\mathcal{A}_2$ that can distinguish between H_2 and H_3. We will construct $\mathcal{A}_1$ that uses $\mathcal{A}_2$ to win the ranked LPN game.

Challenger samples $\mathbf{A} \leftarrow \mathbb{Z}_2^{(N-\ell) \times N}$, $\mathbf{s} \leftarrow \mathbb{Z}_2^{N-\ell}$, and $\mathbf{x} \leftarrow \mathbb{Z}_2^N$ with $wt(\mathbf{x}) = c$. It then sets $\mathbf{e}_0 = \mathbf{A}^\top \mathbf{s} \oplus \mathbf{x}$ and $\mathbf{e}_1 \leftarrow \mathbb{Z}_2^N$. It tosses a bit and sends to $\mathcal{A}_1$, $(\mathbf{A}, \mathbf{e} = \mathbf{e}_b)$. $\mathcal{A}_1$ then generates the corresponding PCM $\mathbf{M}$ for $\mathbf{A}$ and runs $\mathcal{A}_2$ on $\mathbf{Me}$. It is easy to verify that if $b = 0$, $\mathcal{A}_1$ simulates perfectly H_2 and if $b = 1$, it simulates H_3 perfectly. $\mathcal{A}_1$ merely forwards $\mathcal{A}_2$'s guess as its own. This concludes the proof. ◀

Hybrid H_4. $\mathcal{A}$ receives $\mathbf{Me}$ where $\mathbf{M} \leftarrow \mathbb{Z}_2^{\ell \times N}$ and $\mathbf{e} \leftarrow \mathbb{Z}_2^N$.

Note that the difference between hybrids H_3 and H_4 is again the order of sampling. In the former, $\mathbf{A}$ is sampled and then $\mathbf{M}$ is computed, whereas in the latter $\mathbf{M}$ is directly sampled.

Hybrid H_5. $\mathcal{A}$ receives $\mathbf{y} \leftarrow \mathbb{Z}_2^\ell$

Hybrids H_4, H_5 are identically distributed and therefore are statistically indistinguishable. ◀