Provable Time-Memory Trade-Offs: Symmetric Cryptography Against Memory-Bounded Adversaries

Stefano Tessaro and Aishwarya Thiruvengadam

University of California, Santa Barbara
{tessaro,aish}@cs.ucsb.edu

Abstract. We initiate the study of symmetric encryption in a regime where the memory of the adversary is bounded. For a block cipher with n-bit blocks, we present modes of operation for encryption and authentication that guarantee security *beyond* 2^n encrypted/authenticated messages, as long as (1) the adversary's memory is restricted to be less than 2^n bits, and (2) the key of the block cipher is long enough to mitigate memory-less key-search attacks. This is the first proposal of a setting which allows to bypass the 2^n barrier under a reasonable assumption on the adversarial resources.
Motivated by the above, we also discuss the problem of stretching the key of a block cipher in the setting where the memory of the adversary is bounded. We show a tight equivalence between the security of double encryption in the ideal-cipher model and the hardness of a special case of the element distinctness problem, which we call the *list-disjointness problem*. Our result in particular implies a conditional lower bound on time-memory trade-offs to break PRP security of double encryption, assuming optimality of the worst-case complexity of existing algorithms for list disjointness.

Keywords: Foundations, symmetric cryptography, randomness extraction

1 Introduction

Security proofs typically upper bound the maximal achievable advantage of an adversary in compromising a scheme as a function of its *resources*. Almost always, theoretical cryptography measures these resources in terms of *time complexity* – an adversary is considered feasible if its running time is bounded, e.g., by a polynomial, or by some conservative upper bound (e.g., 2^{100}) when the focus is on concrete parameters.

However, time alone does not determine feasibility. Another parameter is the required *memory*. For example, while the naïve birthday attack to find a collision in a hash function with n-bit outputs requires $2^{n/2}$ time *and* memory, well-known collision-finding methods based on Pollard's ρ-method [31] only require $O(n)$

memory. In fact, cryptanalytic attacks often achieve *time-memory* trade-offs, where time complexity increases as the memory usage decreases.

Everything else being equal, we would favor a cryptosystem that requires large memory to be compromised within feasible time over one admitting low-memory attacks. Yet, existing works on provable security that are concerned with adversarial memory costs, such as those dealing with memory-hard functions (e.g., [3,4,6]), consider a more limited scope than the security of classical cryptographic tasks like encryption and authentication. A notable exception is the recent work of Auerbach et al. [7] introducing the concept of a *memory-tight reduction*, which allows lifting conjectured lower bounds on time-memory trade-offs from the underlying assumption to the security of the overall scheme. Fortunately, many reductions are memory-tight, with the exception being mostly reductions in the random-oracle model. This approach, however, still crucially relies on a time-memory assumption for an underlying computational problem, and these are mostly problems studied in public-key cryptography.

This paper: An overview. This paper focuses on *symmetric cryptography* and modes of operation for block ciphers. We present the first schemes for encryption and authentication, based on a block cipher with input length n, that provably achieve security against adversaries which encrypt/authenticate more than 2^n messages, under the assumption that their memory allows storing fewer than 2^n bits. Our results only need fairly standard assumptions (i.e., strong, yet plausible, forms of PRP security) on the underlying block ciphers, and, unlike [7], we only assume hardness with respect to *time*.

Complementary to this, we will discuss how the security of key-length extension methods for block ciphers (and in particular, double encryption) improves under memory restrictions on adversaries, and show conditional results proving optimality of existing attacks against double encryption.

Why this is important. In provably secure symmetric cryptography, the quantity 2^n acts as a barrier on the achievable security in the analysis of schemes based on block ciphers with n-bit inputs, even if the underlying block cipher is very secure (e.g., it is a PRP against adversaries with time complexity 2^{2n}, which is plausible if the key is sufficiently long). The reason is that the core of most proofs is inherently *information-theoretic*, and analyzes the scheme after replacing the block cipher with a truly random permutation (or random function) on n-bit inputs. Here, after $\Omega(2^n)$ queries (either for encryption or verification), the underlying permutation/function is usually queried on *all* inputs – the lack of new randomness breaks down the proof, although the resulting matching attack has often doubly-exponential time complexity in n and it is only a problem because we are considering the (stronger) target of information-theoretic security. For this reason, cryptanalysis often suggests better concrete security guarantees than those given by security proofs. Of course, we have no way to directly deal with time complexity, but here we suggest that bounding the memory of the attacker to be smaller than 2^n can suffice to break this barrier.

Our assumptions. The assumption that attackers have less than 2^n bits of memory is reasonable. While $n = 128$ is common, NSA's Utah data center is

estimated to store 2^{67} bits of data. Moreover, accessing large memory, in practice, adds extra time complexity. Another way to view this is that high security can be achieved *even* when the block size is *smaller*. E.g., we can set $n = 80$ and $k = 128$, and still get beyond 100 bits (i.e., 2^{100} queries) of security.

Note that if we want security against time $T > 2^n$, then we need a security assumption on the block cipher which is true against time-T adversaries. If the key length is larger than $\log(T)$ bits (to thwart the naïve key-search attack), it is not unreasonable to assume that a block cipher is a PRP for T-time attackers, *even* if the block length is n.[1] This however also motivates the general question of what to do if a cipher with longer key does not exist – heuristically, one could use methods for key-length extension [15,21,22,23,24,26,28] that have been validated in the ideal cipher model, and that achieve security against time up to $T = 2^{k+n}$ when the underlying block cipher has key length k. Here, we initiate the study of key-length extension in the memory-bounded setting, and show that, under assumptions we discuss below, key-length extension can be done more efficiently.

1.1 Overview of our Results

We give an overview of the results from this paper. We will start with the case of encryption, before moving to authentication, and our results on key-length extension.

Symmetric encryption. Consider the classical scheme which encrypts each m as $(\mathsf{iv}, \mathsf{E}_K(\mathsf{iv}) \oplus m)$ for a random n-bit iv and a block cipher E with block length n and key K. The canonical $O(2^{n/2})$-query attack against real-or-random (ROR) security waits for two encryptions of m_i and m_j with ciphertexts $c_i = (\mathsf{iv}_i, z_i)$ and $c_j = (\mathsf{iv}_j, z_j)$ such that $\mathsf{iv}_i = \mathsf{iv}_j$, and then checks whether $z_i \oplus z_j = m_i \oplus m_j$. However, if the adversary only has memory to store $O(n \cdot 2^{n/4})$ bits, the attack is not possible, as not all previous ciphertexts can be remembered. The seemingly best-possible strategy is to store $2^{n/4}$ $2n$-bit ciphertexts, and check, for each new query returning $c_i = (\mathsf{iv}_i, z_i)$, whether the iv_i value is used by any of the $2^{n/4}$ ciphertexts, and then proceed as before. This attack however requires $2^{3n/4}$ queries to succeed.

A generalization of the scheme could achieve even higher security: We now pick t random $\mathsf{iv}_1, \ldots, \mathsf{iv}_t$, and the ciphertext is[2]

$$(\mathsf{iv}_1, \ldots, \mathsf{iv}_t, \mathsf{E}_K(\mathsf{iv}_1) \oplus \cdots \oplus \mathsf{E}_K(\mathsf{iv}_t) \oplus m) \ .$$

Of course, we need to *prove* our intuition is valid no matter what a memory-bounded attacker does. We will not be able to do so for this specific scheme, but consider a related scheme, which we call *sample-then-extract*, using an extractor $\mathsf{Ext} : \{0,1\}^{n \cdot t} \times \{0,1\}^s \to \{0,1\}^\ell$ to encrypt an ℓ-bit message as

$$(\mathsf{iv}_1, \ldots, \mathsf{iv}_t, \mathsf{seed}, \mathsf{Ext}(\mathsf{E}_K(\mathsf{iv}_1) \| \cdots \| \mathsf{E}_K(\mathsf{iv}_t), \mathsf{seed}) \oplus m) \ ,$$

[1] For example, an ideal cipher with key length $\log(T)$ *is* a PRP against T-time attackers.

[2] This scheme was proposed in [13], with the different purpose of proving security beyond the birthday bound.

where $\mathsf{seed} \xleftarrow{\$} \{0,1\}^s$ is chosen randomly upon each encryption.

For example, assuming Ext is a sufficiently strong extractor, $\ell = n$, $t = 32n$, we will show security up to $q = 2^{1.5n}$ encryption queries for attackers with running time $T \geq q$ and memory $S \leq 2^{n(1-o(1))}$, provided E is secure against T-time attackers as a PRP.

The connection with sub-key prediction. Our proof relies on the problem of *sub-key prediction*, which was recently revisited [11,14] in the context of big-key encryption, but which initially appeared implicitly in previous entropy preservation lemmas [5,30,36].[3] In particular, the core of the proof involves a hybrid world where the block cipher E_K is replaced by a random permutation P. For every i, we imagine an experiment where we run the attacker for the first $i-1$ queries, all answered using the encryption scheme with P in lieu of E_K, and then look at its S-bit state σ_{i-1} before it makes the i-th query. Then, we know that the average-case min-entropy of the *permutation* P given σ_{i-1} is at most S bits lower than the maximum, i.e., $\log(2^n!) \approx n \cdot 2^n$. The existing bounds on sub-key prediction give us directly a lower bound on the min-entropy of $P(\mathsf{iv}_1)\| \cdots \|P(\mathsf{iv}_t)$ conditioned on σ_{i-1}. If Ext is a suitable extractor, this makes its output random, and thus this masks the ciphertext.

The proof is perhaps obvious in retrospect, but it highlights a few interesting traits: First off, the idea of a reduction to sub-key prediction is novel. Second, handling random permutations (vs functions) comes for free by simply considering a different entropy lower bound for which the extractor needs to work.

Authentication. The next logical step is to build a *message authentication code* (MAC) for ℓ-bit messages from an n-bit block cipher, with security for $q > 2^n$ queries for adversaries with memory $S < 2^n$. Here, $\ell > n$ in order for the question to make sense. This appears harder – as we will explain in the body in detail, if we want to go as far as building a PRF (as it is usually the case when proving security of MAC constructions), the resulting construction is likely to yield (at least when following the canonical proof approach) a PRG which is unconditionally secure for unrestricted[4] space-bounded branching programs, with much better stretch than the existing state-of-the-art [16,27], and this is currently out of reach.

We overcome this by considering a (minimally) *interactive* approach to the problem of message authentication, which we refer to as *synchronous authentication.* In this setting, we force the output of the MAC to also depend on a random challenge previously sent by the other party. For example, whenever Alice sends an authenticated message to Bob, she also sends a challenge to be used by Bob to authenticate his next message to Alice. Our construction makes t calls per *bit* of the message, for a parameter t.[5] In particular, a challenge consists of t n-bit strings $\mathsf{iv}_1, \ldots, \mathsf{iv}_t$, as well as an extractor seed seed. Then, the tag of a message

[3] In fact, the simplest lemma by Alwen, Dodis, and Wichs [5] will suffice for our purposes. One could likely obtain better concrete bounds using the techniques from [11], yet their bounds are hard to express explicitly, and we do not explore this route here.

[4] I.e., they can learn the output bits of the PRG adaptively, with no restrictions.

[5] A higher-rate version of the scheme can be given, at the price of lower security.

$M = M_1 M_2 \dots M_\ell \in \{0,1\}^\ell$ is obtained by computing the values

$$Y_i = E_K(\langle i \rangle \| M_i \| \mathsf{iv}_1) \| \cdots \| E_K(\langle i \rangle \| M_i \| \mathsf{iv}_t) \ ,$$

where $\langle i \rangle$ is a $\log \ell$-bit encoding of i, and finally outputting the message tag $T = \bigoplus_{i=1}^{t} \mathsf{Ext}(Y_i, \mathsf{seed})$, where Ext is a randomness extractor.

We introduce a definition of synchronous message authentication and prove our scheme secure. Again, our proof will resort to a reduction to the unpredictability of the Y_i values via sub-key prediction, but an extra complication is that we need to analyze a more complex security game than in the case of encryption, where the adversary can authenticate *adaptively* chosen messages.

The block cipher assumption and double encryption. If we want to prove security beyond 2^n queries, we need to use a block cipher whose PRP security holds for an attacker which runs for time $T \geq 2^n$ time and has memory $S \ll 2^n$. But: *What should we do when the key is not long enough?*

We can of course always extend the length of a key to a block cipher by using conventional key-length extension methods which are validated in the ideal-cipher model [15,21,22,23,24,26,28]. One observation however is that if we are assuming a bound on the adversary's memory, one could achieve better security and/or better efficiency (for comparable security). To this end, we initiate the study of key-length extension in the memory-bounded regime.

In particular, we look at *double encryption* (DE), i.e., given a block cipher E, we consider a new block cipher that uses two keys K_1, K_2 to map x to $\mathsf{E}_{K_1}(\mathsf{E}_{K_2}(x))$. The best known attack against DE achieves a time-memory trade-off[6] of $T^2 \cdot S = 2^{3k}$ with $T \geq 2^k$ – this was first pointed out in the work of van Oorschot and Wiener [38]. If one can show that this is indeed optimal, then we can for example hope to achieve security against time $T = 2^{1.25k}$ when $S \ll 2^{0.5k}$. In other words, in contrast to common wisdom, double encryption would increase security if memory is bounded.

Verifying this unconditionally, while possible (recall we are content with a proof in the ideal-cipher model), appears to be out of reach. However, we establish a connection between the PRP security of DE in the ICM and a problem we call *list disjointness*. In this problem, we assume we are given *two* equally long lists L_1 and L_2 as inputs, each of distinct elements, with the promise that either (1) $L_1 \cap L_2 = \emptyset$ or (2) $|L_1 \cap L_2| = 1$. An algorithm is given access to the lists as an oracle (i.e., for an i and b, it can obtain the i-th element of L_b), and the goal is to assess whether (1) or (2) holds. This problem is a special case of the well-known *element distinctness* problem [17,40], where the algorithm is given oracle access to a single list L and needs to decide whether its elements are distinct. In particular, every algorithm for distinctness yields one for list disjointness, by letting L be the concatenation of L_1 and L_2.

It is not hard to see that every algorithm for list disjointness yields a PRP distinguisher for DE with similar query and memory complexities. More interestingly, we also show that every PRP distinguisher for DE yields an algorithm

[6] For comparison, the textbook meet-in-the-middle attack achieves a tradeoff of $T \cdot S = 2^{2k}$.

(with similar query and memory complexities) that solves list disjointness in *the worst case*.

First off, there has been little progress in providing general lower bounds for query-memory trade-offs for element distinctness (existing lower bounds consider either restricted algorithms [40], and can be bypassed by more general algorithms [8], or are far from known upper bounds [2,9]). The situation does not appear easier for list disjointness. Progress on proving a tight lower bound for query-memory trade-offs for the PRP security seems therefore to necessarily involve new non-trivial insights.

Second, and perhaps more interestingly, the best algorithm for element distinctness is due to Beame, Clifford, and Machmouchi [8], and achieves a tradeoff of $T^2 \cdot S = |L|^3$. The algorithm also applies to list disjointness, and assuming it is optimal, by our reduction we get a conditional lower bound confirming the best-known time-memory trade-off for DE to be optimal.

1.2 Further Related Works

The bulk of the interest on bounded-memory algorithms stems from complexity theory. In particular, a number of works have been concerned with lower bounds for time-memory trade-offs in restricted complexity classes, such as pebbling models and branching programs. Textbooks like that of Savage [35] provide a comprehensive introduction to the topic. Particularly relevant to us is the work on building PRGs for space-bounded computation [29], which was the first to show unconditional pseudorandomness for space-bounded distinguishers.

Our work is also very related to that of Raz [32,33] on time-memory trade-offs for learning parities (and related problems). Raz shows in particular an encryption scheme with an n-bit key which unconditionally resists an attacker with memory smaller than n^2/c for a constant c when encrypting an exponential number of plaintexts. Our encryption scheme can be seen as replacing the n-bit key with a much larger random permutation table. Raz's technique is not applicable because it would require evaluating the permutation at $\Theta(2^n)$ positions upon each encryption. Time-memory trade-offs for learning lower-weight parities were also given [20], but it does not appear possible to exploit these results to obtain a cryptosystem.

Outline of this paper. Section 2 will introduce technical tools needed throughout the paper, including our model of computation, information-theoretic preliminaries, and notation for the sub-key prediction problem. Sections 3 and 4 discuss our encryption and authentication schemes. Section 5 presents our results on double encryption.

2 Preliminaries

Throughout this paper, let $N = 2^n$ for an understood $n \in \mathbb{N}$. Also, let $[i]$ denote the set $\{1, 2, \ldots, i\}$. As usual, we use the notation $|r|$ to denote the

length of string r in bits. By $r \xleftarrow{\$} \{0,1\}^n$, we indicate that r is chosen uniformly from $\{0,1\}^n$. We let $\mathcal{F}_{m,n}$ denote the uniform distribution over functions from $\{0,1\}^m$ to $\{0,1\}^n$ and let $\mathcal{P}_n$ denote the uniform distribution over permutations on $\{0,1\}^n$. We also write $\mathcal{F}$ and $\mathcal{P}$ for $\mathcal{F}_{n,n}$ and $\mathcal{P}_n$ whenever n is clear from the context.

2.1 Information-theoretic Preliminaries

The *min-entropy* of a random variable X (taking values from a set $\mathcal{X}$) is $\mathrm{H}_\infty(X) = -\min_{x\in\mathcal{X}} \log(\mathsf{Pr}[X=x])$. Moreover, for two jointly distributed random variables X, Y, and an element y such that $\mathsf{Pr}[Y=y] > 0$, we define $\mathrm{H}_\infty(X|Y=y) = \min_{x\in\mathcal{X}} \log\left(1/\mathsf{Pr}\left[X=x \,\middle|\, Y=y\right]\right)$. This is in particular the conditional min-entropy conditioned on a particular *outcome*. When conditioning on a random variable, we use the *average-case* version of min-entropy [19], i.e.,

$$\mathrm{H}_\infty(X|Y) = -\log\left(\sum_{y\in\mathcal{Y}} \max_{x\in\mathcal{X}} \mathsf{Pr}[X=x, Y=y]\right) .$$

We will need the following simple fact about average-case min-entropies.

Lemma 1 ([19]). *Let X, Y, Z be random variables. If Y can take at most 2^λ values, then*

$$\mathrm{H}_\infty(X|YZ) \geq \mathrm{H}_\infty(XY|Z) - \lambda \geq \mathrm{H}_\infty(X|Z) - \lambda . \tag{1}$$

Extractors. Recall that a function $\mathsf{Ext} : \{0,1\}^{t\cdot n} \times \{0,1\}^s \to \{0,1\}^\ell$ is said to be a (γ,ε)-strong extractor if for every random variable X on $\{0,1\}^{t\cdot n}$ with $\mathrm{H}_\infty(X) \geq \gamma$, $(U_s, \mathsf{Ext}(X,U_s))$ is ε-close to (U_s, U_ℓ). We say that $H : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^\ell$ is 2-universal if for all n-bit $x \neq x'$, we have $\mathsf{Pr}[K \xleftarrow{\$} \{0,1\}^k : H(K,x) = H(K,x')] = 2^{-\ell}$. The following is well known.

Lemma 2 (Leftover Hash Lemma). *[25] If $H : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^\ell$ is 2-universal, and $\ell = \gamma - 2\log(1/\varepsilon)$, then $\mathsf{Ext}(x,K) := H(K,x)$ is a strong (γ,ε)-extractor.*

Following Dodis et al. [19], we also say that $\mathsf{Ext} : \{0,1\}^{t\cdot n} \times \{0,1\}^s \to \{0,1\}^\ell$ is an *average-case* (γ,ε)-strong extractor if for all pairs of random variables (X,I) such that X in $\{0,1\}^{t\cdot n}$ satisfies $\mathrm{H}_\infty(X|I) \geq \gamma$, $(U_s, \mathsf{Ext}(X,U_s), I)$ is ε-close to (U_s, U_ℓ, I).

In [19] the leftover hash lemma is extended to show that universal hash functions yield an average-case strong extractor with the same parameters. In general, with a slight loss in parameters, a (γ,ε)-(strong) extractor is also an average-case $(\gamma, 3\varepsilon)$-(strong) extractor as stated as shown by [37].

Entropy Preservation. Assume we are given a vector $X \in (\{0,1\}^m)^N$, which we often will think of as the table of a function $[N] \to \{0,1\}^m$. Further, let us sample indices $i_1, \ldots, i_t$ uniformly at random from $[N]$, and consider the induced random variable

$$X[i_1, \ldots, i_t] = X_{i_1}, \ldots, X_{i_t} \ .$$

We are interested in the relationship between the entropy of X and that of $X[i_1, \ldots, i_t]$. The following lemma was proven by Alwen, Dodis, and Wichs [5], and considers the more general setting where we are given some auxiliary information Z, and the indices $i_1, \ldots, i_t$ are sampled independently of X and Z.[7]

Lemma 3. *Let (X, Z) be correlated random variables, where $X \in (\{0,1\}^m)^N$, and $I = (i_1, \ldots, i_t) \xleftarrow{\$} [N]^t$. Further, assume that $\mathrm{H}_\infty(X|Z) \geq N(m-1) - L$, where $L \leq (1-\delta)Nm$ for some $\delta \in [0,1]$. Then, $\mathrm{H}_\infty(X[I]|Z, I) \geq \gamma$, if*

$$\delta \geq \left[\frac{2\gamma}{t} \left(1 + \frac{n}{m}\right) + \frac{1}{m} + \frac{3\gamma + 5}{Nm} \right] .$$

Note that for our application scenarios, $\left(1 + \frac{n}{m}\right) \approx 2$ and $\frac{3\gamma+5}{Nm} \to 0$, so this means in particular that we get γ bits of entropy for every $\gamma \leq t(\delta - 1/m)/4$.

2.2 Model of Computation and Cryptographic Primitives

We will consider a model of computation with space-bounded adversaries, inspired by the one from [4,6]. In particular, we consider adversaries $\mathcal{A}$ making queries to an oracle $\mathcal{O}$. This accommodates without loss of generality for the case where $\mathcal{A}$ makes queries to *multiple* oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$, which we view as one individual oracle with an appropriate addressing input. We will not specify the model of execution of $\mathcal{A}$ any further at the lowest level of detail (but we assume we fix one specific model of computation), but will introduce some convenient relaxation of memory-bounded executions that will suffice for our purposes.

More specifically, the execution of an adversary proceeds in *stages* (or *steps*), allowing one oracle query in each stage. In particular, the execution of $\mathcal{A}$ starts with the state $\sigma_0 = x$, where x is the input, and no previous-query answer $y_0 = \perp$. Then, in the i-th stage, the adversary computes, as a function of the state σ_{i-1} and the previous query answer y_{i-1}, a query q_i to $\mathcal{O}$, as well as the next state σ_i. Thus, formally, an adversary $\mathcal{A}$ is a randomized algorithm implementing a map $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$. In most proofs, we will generally not need to restrict the actual space complexity of $\mathcal{A}$ itself, as long as the states σ_i are bounded in size.

We say that an adversary $\mathcal{A}$ is *S-bounded* if $|\sigma_i| \leq S$ holds for all states in the execution. We further say that an adversary $\mathcal{A}$ has time complexity (or running time) T if an execution takes overall at most T steps. We say it has *(description) size D* if the description of $\mathcal{A}$ requires at most D bits. Finally, it makes q queries if it takes q steps, resulting in q queries to $\mathcal{O}$.

[7] We note that Lemma 3 has a different expression for δ than what would be implied by the original statement [5, Lemma A.3], but this is due to a missing factor of $\frac{2\gamma}{t}$ in one of the terms (which can be inferred from their proof).

BLOCK CIPHERS AND PRPS. A *block cipher* is a function $\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, where $\mathsf{E}_K = \mathsf{E}(K,\cdot)$ is a permutation for all $K \in \{0,1\}^k$. Generally, we assume that E is efficiently computable *and* invertible.

We define PRP security in terms of the *PRP-CPA-advantage* of an adversary $\mathcal{A}$ against a block cipher E, which is

$$\mathsf{Adv}^{\mathsf{PRP\text{-}CPA}}_{\mathsf{E}}(\mathcal{A}) = \left|\mathsf{Pr}[K \xleftarrow{\$} \{0,1\}^k : \mathcal{A}^{\mathsf{E}_K} = 1] - \mathsf{Pr}[P \xleftarrow{\$} \mathcal{P}_n : \mathcal{A}^P = 1]\right| .$$

We also define $\mathsf{Adv}^{\mathsf{PRP\text{-}CPA}}_{\mathsf{E}}(D,T,q,S) = \max_{\mathcal{A}}\{\mathsf{Adv}^{\mathsf{PRP\text{-}CPA}}_{\mathsf{E}}(\mathcal{A})\}$, where the maximum is taken over all S-bounded adversaries $\mathcal{A}$ that run in time at most T, making q queries at most, and with size at most D.

Note that PRP security does not need to depend on the block length n if the key is long enough. Below, we repeatedly make the assumption that there exist block ciphers $\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ which are secure PRPs for time complexities $T > 2^n$ (and suitably small size D) and space complexity $S < 2^n$. Note that this implicitly implies $k(n) > \log T$. This is easily seen to be satisfied by an ideal cipher, even if S is unbounded.

2.3 Sub-key Prediction

In the sub-key prediction problem [11,14], the adversary $\mathcal{A}$ is given some leakage σ on a *key*, which here we interpret as a function $F : \{0,1\}^n \to \{0,1\}^n$. The leakage is derived through some (adversarially chosen) function $\mathcal{L}$. Then, for randomly chosen indices $i_1, \ldots, i_t$, $\mathcal{A}$ tries to guess the "sub-key" $K = F(i_1) \,\|\, \ldots \,\|\, F(i_t)$, i.e., the evaluations of the function at those indices. We generalize this notion further by allowing for auxiliary information Z correlated with F. In particular, we allow both $\mathcal{L}$ and $\mathcal{A}$ to access Z. (Still, we will omit Z when not necessary.)

More formally, we consider an adversary $\mathcal{A}$ with leakage function $\mathcal{L}$ interacting in the game $G^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(\mathcal{A},\mathcal{L})$ described in Figure 1. Here, we stress that both $\mathcal{A}$ and $\mathcal{L}$ are computationally unbounded with no limits on their memory–the only limitation is the size of σ. The game is parameterized by the distribution $\mathcal{D}$ according to which (F,Z) are chosen, the distribution $\mathcal{I}$ according to which the indices are chosen, and the number of indices t.

We can then define advantage measures for an adversary in guessing the sub-key correctly in the game $G^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(\mathcal{A},\mathcal{L})$ as follows.

Definition 1. *The advantage of an adversary $\mathcal{A}$ with leakage function $\mathcal{L}$ in the game $G^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(\mathcal{A},\mathcal{L})$ is defined as*

$$\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(\mathcal{A},\mathcal{L}) = \mathsf{Pr}[G^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(\mathcal{A},\mathcal{L}) = \mathsf{true}] .$$

Furthermore, we define

$$\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(S) = \max_{\mathcal{L}:\mathcal{D}\to\{0,1\}^S} \max_{\mathcal{A}}\{\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(\mathcal{A},\mathcal{L})\} .$$

Game $G^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(\mathcal{A},\mathcal{L})$:

$(F, Z) \xleftarrow{\$} \mathcal{D}$; $\sigma \leftarrow \mathcal{L}(F, Z)$
$(i_1, \ldots, i_t) \xleftarrow{\$} \mathcal{I}$; $K \xleftarrow{\$} \mathcal{A}(\sigma, Z, i_1, \ldots, i_t)$
Return $(K = F(i_1) \| \ldots \| F(i_t))$

Fig. 1: **Game** $G^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(\mathcal{A},\mathcal{L})$**.** Game defining sub-key prediction with auxiliary information. The adversary, given leakage σ and auxiliary information Z on F, wins if it guesses the output of F at indices $i_1, \ldots, i_t$.

Often $\mathcal{I}$ will be the uniform distribution over t-tuples of indices in $(\{0,1\}^n)^t$, for notational convenience, we drop the subscript $\mathcal{I}$ and simply refer to the advantage as $\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D},t}(S)$ in such cases.

The following lemma is immediate by definition of conditional min-entropy.

Lemma 4. *If* $\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D},\mathcal{I},t}(S) \leq 2^{-\gamma}$*, then for* $(F, Z) \xleftarrow{\$} \mathcal{D}$*,* $(\mathsf{iv}_1, \ldots, \mathsf{iv}_t) \xleftarrow{\$} \mathcal{I}$ *and* $\sigma \leftarrow \mathcal{L}(F, Z)$*, we have*

$$\mathrm{H}_\infty\left(F(\mathsf{iv}_1) \| \ldots \| F(\mathsf{iv}_t) | \sigma, (\mathsf{iv}_1, \ldots \mathsf{iv}_t), Z\right) \geq \gamma \ .$$

We now derive the advantage of an adversary in the sub-key prediction game with auxiliary information when the leakage function outputs exactly S bits. In particular, the following lemma is a straightforward application of Lemmas 1 and 3.

Lemma 5 (Sub-key Prediction with Auxiliary Information). *Let correlated random variables* (F, Z) *be chosen according to a distribution* $\mathcal{D}$ *such that* $F : \{0,1\}^n \to \{0,1\}^n$ *and* $\mathrm{H}_\infty(F|Z) \geq N(n-1) - L$*.*

Let $S + L \leq (1-\delta)nN$ *for some* $\delta \in [0,1]$*. Then,* $\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D},t}(S) \leq 2^{-\gamma}$ *if* $\delta \geq \left[\frac{4\gamma}{t} + \frac{1}{n} + \frac{3\gamma+5}{nN}\right]$*.*

In comparison to [5], the recent work by Bellare and Dai [11] provides better concrete bounds for sub-key prediction in the case where F is uniformly distributed over all functions, and with no auxiliary information (or, more generally, Z is independent of F). However, we use [5] as we need to handle both auxiliary information and the case that F is a permutation. Also, while it may be possible to extend the proofs of [11] to this more general setting, the resulting bounds are hard to express analytically. Either way, our results are generic and an improvement on sub-key prediction bounds will directly yield better bounds for our instantiations below.

3 Encryption

We give an encryption scheme for which the amount of time needed to break it increases as the memory of the adversary decreases, in particular going beyond

Game $\mathsf{ROR}^{\mathcal{E},b}(\mathcal{A})$:	Oracle $\mathcal{E}'(M,b)$:
$K \xleftarrow{\$} \mathsf{Gen}$	If $b = 0$ then return $c \xleftarrow{\$} \mathsf{Enc}_K(M)$
$b' \xleftarrow{\$} \mathcal{A}^{\mathcal{E}'(\cdot,b)}$	If $b = 1$ then
Return b'	choose $M' \xleftarrow{\$} \mathcal{M}$ such that $\lvert M'\rvert = \lvert M\rvert$
	Return $c \xleftarrow{\$} \mathsf{Enc}_K(M')$.

Fig. 2: **Game** $\mathsf{ROR}^{\mathcal{E},b}(\mathcal{A})$**.** Game defining the real-or-random security of the encryption scheme $\mathcal{E}$, where $b \in \{0,1\}$.

2^n, where n is the block length of an underlying block cipher. To this end, we first recall the standard definition of a symmetric-key encryption scheme, its security, and introduce some additional notational conventions.

Encryption Scheme: Syntax. An *encryption scheme* is a tuple of algorithms $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ where: (1) the *key generation algorithm* Gen outputs a key K, (2) the *encryption algorithm* Enc takes as input the secret key K and a message M (from some understood message space $\mathcal{M}$), and outputs a ciphertext $c \xleftarrow{\$} \mathsf{Enc}_K(M)$, and (3) the *decryption algorithm* Dec takes as input the secret key K and a ciphertext c and outputs a message $M \leftarrow \mathsf{Dec}_K(c)$. The correctness requirement is that for any key K output by Gen, and message $M \in \mathcal{M}$, we have $\mathsf{Dec}_K(\mathsf{Enc}_K(M)) = M$ with large probability (usually one).

Occasionally, it will be convenient to think of the key K as a function $F : \{0,1\}^n \to \{0,1\}^n$ (to be instantiated for example with a block cipher), to which the scheme is given oracle access. In this case, we will simply write Enc^F and Dec^F instead of Enc_K and Dec_K. Then one can get for example $\mathsf{Enc}_K = \mathsf{Enc}^{\mathsf{E}_K}$ for the final block cipher instantiation.

Security of Encryption Schemes. We briefly review the notion of *real-or-random (ROR)* security [12] of an encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$: we consider the games $\mathsf{ROR}^{\mathcal{E},b}(\mathcal{A})$ (for $b \in \{0,1\}$) for an adversary $\mathcal{A}$, as described in Figure 2, and define

$$\mathsf{Adv}^{\mathsf{ROR}}_{\mathcal{E}}(\mathcal{A}) = \left|\mathsf{Pr}[\mathsf{ROR}^{\mathcal{E},0}(\mathcal{A}) = 1] - \mathsf{Pr}[\mathsf{ROR}^{\mathcal{E},1}(\mathcal{A}) = 1]\right| ,$$

as well as $\mathsf{Adv}^{\mathsf{ROR}}_{\mathcal{E}}(D,T,q,S) = \max_{\mathcal{A}}\{\mathsf{Adv}^{\mathsf{ROR}}_{\mathcal{E}}(\mathcal{A})\}$, where the maximum is taken over all S-bounded adversaries $\mathcal{A}$ with running time at most T, making at most q queries, and have size at most D.

For our intermediate information-theoretic steps below, our statements will not depend on D and T, and we simply write $\mathsf{Adv}^{\mathsf{ROR}}_{\mathcal{E}}(q,S) = \mathsf{Adv}^{\mathsf{ROR}}_{\mathcal{E}}(\infty,\infty,q,S)$.

3.1 The Sample-then-Extract Scheme

The scheme is best described using a distribution $\mathcal{D}$ on functions from n bits to n bits as a parameter. In addition, let $\mathsf{Ext} : \{0,1\}^{tn} \times \{0,1\}^s \to \{0,1\}^\ell$, and let $\mathcal{I}$

be the uniform distribution over $\{0,1\}^{tn}$. The encryption scheme $\mathsf{StE}[\mathcal{D}, t, \mathsf{Ext}] = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ for messages in $\mathcal{M} = \{0,1\}^{\ell}$ is then defined as follows:

Scheme $\mathsf{StE}[\mathcal{D}, t, \mathsf{Ext}]$:

- **Key generation.** The key generation algorithm Gen outputs $F \xleftarrow{\$} \mathcal{D}$, where $F : \{0,1\}^n \to \{0,1\}^n$.
- **Encryption.** On input $M \in \mathcal{M}$, Enc^F does the following:
  1. $\mathsf{seed} \xleftarrow{\$} \{0,1\}^s$.
  2. $\mathsf{iv} = (\mathsf{iv}_1, \ldots, \mathsf{iv}_t) \xleftarrow{\$} \mathcal{I}$.
  3. $c \leftarrow \mathsf{Ext}(F(\mathsf{iv}_1) \| \ldots \| F(\mathsf{iv}_t), \mathsf{seed}) \oplus M$
  4. Return $(c, \mathsf{seed}, \mathsf{iv}_1, \ldots, \mathsf{iv}_t)$.
- **Decryption.** On input $(c, \mathsf{seed}, \mathsf{iv}_1, \ldots, \mathsf{iv}_t)$, Dec computes $M \leftarrow (\mathsf{Ext}(F(\mathsf{iv}_1) \| \ldots \| F(\mathsf{iv}_t), \mathsf{seed})) \oplus c$, and returns M.

We will then instantiate our scheme with a block cipher E, and in this case we refer to the scheme as $\mathsf{StE}[\mathsf{E}, t, \mathsf{Ext}]$. This is the special case of the above scheme when the distribution $\mathcal{D}$ samples the function $\mathsf{E}_K(\cdot)$ for $K \xleftarrow{\$} \{0,1\}^k$ where k is the key-length of E.

3.2 Security of StE

We now prove the security of StE. Our main theorem is in the information-theoretic setting, where we reduce security to the sub-key prediction problem for the distribution $\mathcal{D}$. Then, below, we instantiate the scheme with a block cipher E, assumed to be a PRP, and use the theorem to give corresponding security statements for this instantiation, showing in particular we can attain security beyond 2^n queries.

Theorem 1 (Information-theoretic security of StE). *Assume that*

$$\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D},t}(S + s + \ell + tn) \leq 2^{-\gamma}$$

and that $\mathsf{Ext} : \{0,1\}^{tn} \times \{0,1\}^s \to \{0,1\}^{\ell}$ *is an average-case* (γ, ε)*-strong extractor. Then,*

$$\mathsf{Adv}^{\mathsf{ROR}}_{\mathsf{StE}[\mathcal{D},t,\mathsf{Ext}]}(q, S) \leq q\varepsilon \ .$$

Proof. The proof proceeds in two parts. In the first part, we consider a variant of the sub-key prediction problem where the adversary, instead of trying to predict the sub-key at the given indices predicts, whether it has received the output of an extractor applied to the sub-key or a uniform random string. More precisely, consider a pair of adversaries $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ where $\mathcal{A}'_1$ outputs $S + s + \ell + tn$ bits, and define the game $G^b(\mathcal{A}')$ as follows:

- $F \xleftarrow{\$} \mathcal{D}$; $\sigma \leftarrow \mathcal{A}'_1(F)$; $\mathsf{iv} = (\mathsf{iv}_1, \ldots, \mathsf{iv}_t) \xleftarrow{\$} \{0,1\}^{tn}$; $\mathsf{seed} \xleftarrow{\$} \{0,1\}^s$
- If $b = 0$ then $c \leftarrow \mathsf{Ext}(F(\mathsf{iv}_1) \| \ldots \| F(\mathsf{iv}_t), \mathsf{seed})$

- If $b = 1$ then $c \xleftarrow{\$} \{0,1\}^\ell$
- $b' \leftarrow \mathcal{A}'_2(\sigma, c, \mathsf{seed}, \mathsf{iv}_1, \ldots, \mathsf{iv}_t)$
- Return b'

The following lemma bounds is a simple corollary of Lemma 4 and the fact that Ext is an average-case (γ, ε)-strong extractor.

Lemma 6. *If* $\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D},t}(S + \ell + s + tn) \leq 2^{-\gamma}$ *and* $\mathsf{Ext} : \{0,1\}^{tn} \times \{0,1\}^s \to \{0,1\}^\ell$ *is an average-case* (γ, ε)*-strong extractor, then*

$$\left|\mathsf{Pr}[G^0(\mathcal{A}') = 1] - \mathsf{Pr}[G^1(\mathcal{A}') = 1]\right| \leq \varepsilon \ .$$

We now introduce hybrids H_i for $i = 0, \ldots, q$ such that in hybrid experiment i-th hybrid, the adversary $\mathcal{A}$ interacts with the oracle $\mathcal{E}'(M, 0)$ for the first i queries and with $\mathcal{E}'(M, 1)$ for the remaining queries. Formally, for $i = 1, \ldots, q$, we define the following hybrid experiment $H^{\mathsf{StE}}_i(\mathcal{A})$ for an adversary $\mathcal{A}$:

$$F \xleftarrow{\$} \mathsf{Gen};\ b' \leftarrow \mathcal{A}^{\mathcal{E}'(\cdot,i)};\ \text{Return } b'$$

where $\mathcal{E}'(M, i)$ responds to the j-th query as follows:

- If $j \leq i$, return $c \xleftarrow{\$} \mathsf{Enc}^F(M)$.
- Else, choose $M' \xleftarrow{\$} \mathcal{M}$ such that $|M'| = |M|$ and return $c \xleftarrow{\$} \mathsf{Enc}^F(M')$.

Then, by definition of the advantage $\mathsf{Adv}^{\mathsf{ROR}}_{\mathcal{E}}(\mathcal{A})$, we have

$$\mathsf{Adv}^{\mathsf{ROR}}_{\mathcal{E}}(\mathcal{A}) = \left|\mathsf{Pr}[H^{\mathsf{StE}}_q(\mathcal{A}) = 1] - \mathsf{Pr}[H^{\mathsf{StE}}_0(\mathcal{A}) = 1]\right| \ . \tag{2}$$

We now prove the following central lemma.

Lemma 7. $|\mathsf{Pr}[H^{\mathsf{StE}}_i(\mathcal{A}) = 1] - \mathsf{Pr}[H^{\mathsf{StE}}_{i-1}(\mathcal{A}) = 1]| \leq \varepsilon$.

Proof. We now construct an adversary $\mathcal{A}' = (\mathcal{A}'_1, \mathcal{A}'_2)$ for the game $G^b(\mathcal{A}')$ introduced earlier. On input F, $\mathcal{A}'_1$ proceeds as follows:

- $(\sigma_0, y_0) \leftarrow \perp$
- for $j = 1$ to $i - 1$
  - $\star$ $(M_j, \sigma_j) \leftarrow \mathcal{A}(\sigma_{j-1}, y_{j-1})$
  - $\star$ $y_j \leftarrow \mathsf{Enc}^F(M_j)$
- Return (σ_{i-1}, y_{i-1})

Note that the output length of $\mathcal{A}'_1$ is at most S plus the length of a ciphertext, i.e., $S + s + \ell + n \cdot t$.

Now, the adversary $\mathcal{A}'_2$, is given (σ_{i-1}, y_{i-1}) from $\mathcal{A}'_1(F)$, and moreover, it receives $(u, \mathsf{seed}, \mathsf{iv}_1, \ldots, \mathsf{iv}_t)$ as its challenge from the game. It then proceeds as follows: it continues the execution of $\mathcal{A}$ with input (σ_{i-1}, y_{i-1}) and when $\mathcal{A}$ makes its i-th query by requesting the encryption of a message M, the adversary $\mathcal{A}'_2$ answers this query to $\mathcal{A}$ with the ciphertext $(u \oplus M, \mathsf{seed}, \mathsf{iv}_1, \ldots, \mathsf{iv}_t)$. It then continues the execution of $\mathcal{A}$, but answers all future encryption queries with truly random ciphertexts.

By construction, we now have

$$|\mathsf{Pr}[\overline{H}_i^{\mathsf{StE}}(\mathcal{A}) = 1] - \mathsf{Pr}[\overline{H}_{i-1}^{\mathsf{StE}}(\mathcal{A}) = 1]| = |\mathsf{Pr}[G^0(\mathcal{A}') = 1] - \mathsf{Pr}[G^1(\mathcal{A}') = 1]|$$

Applying Lemma 6 then concludes the proof of the lemma. □

Thus, Equation 2 and Lemma 7 yield

$$\mathsf{Adv}^{\mathsf{ROR}}_{\mathcal{E}}(\mathcal{A}) \leq \sum_{i=1}^{q} \left|\mathsf{Pr}[H_i^{\mathsf{StE}}(\mathcal{A}) = 1] - \mathsf{Pr}[H_{i-1}^{\mathsf{StE}}(\mathcal{A}) = 1]\right| \leq q \cdot \varepsilon \ ,$$

which gives us the theorem. □

Instantiation. We now derive a corollary stating the security of the encryption scheme with a block cipher $\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ assumed to be a good pseudorandom permutation (PRP). We instantiate the extractor in the encryption scheme using the leftover hash lemma (cf. Lemma 2). The following lemma follows by replacing the block cipher with a randomly chosen permutation F (at the cost of the PRP advantage), and then using the fact that F has min-entropy $\log(N!)$.

Corollary 1 (Instantiation of StE). *Let $\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Let $\mathcal{H} : \{0,1\}^{tn} \times \{0,1\}^{tn} \to \{0,1\}^\ell$ be a 2-universal family of hash functions. Let $S \leq (1-\delta)nN$ for some $\delta \in [0,1]$. Then, if $\delta \geq \left\lceil \frac{4(\ell - 2\log\varepsilon)}{t} + \frac{1}{n} + \frac{4\ell - 6\log\varepsilon + 2tn + 5}{nN} \right\rceil$ for some $\varepsilon > 0$, then for all D, T, there exists $D' \approx D$ and $T' \approx T$ such that*

$$\mathsf{Adv}^{\mathsf{ROR}}_{\mathsf{StE}[\mathsf{E},t,\mathcal{H}]}(D,T,q,S) \leq q\varepsilon + \mathsf{Adv}^{\mathsf{PRP\text{-}CPA}}_{\mathsf{E}}(D',T',tq,S+2n(t-1)) \ .$$

Beyond 2^n-security. We plug in concrete values in Corollary 1 to demonstrate that our encryption scheme can tolerate $q \gg 2^n$ queries by the adversary, as long as memory is bounded.

With $N = 2^n$, let $q \leq N^{1.5}$ and we want ε to be 2^{-3n} such that in particular $q\varepsilon \leq 2^{-1.5n}$ for an S-bounded adversary where $S \leq N^{1-\alpha}$ with $0 < \alpha \ll 1$. If $\ell = n$ and $t = an$ where $n \geq 20$ and $a \geq 32$, we have

$$\mathsf{Adv}^{\mathsf{ROR}}_{\mathsf{StE}[\mathsf{E},t,\mathcal{H}]}(D,T,q,S) \leq 2^{-1.5n} + \mathsf{Adv}^{\mathsf{PRP\text{-}CPA}}_{\mathsf{E}}(D',T',tq,S+2n(t-1)) \ .$$

As for the PRP-advantage term, it is reasonable to assume for a good block cipher, the advantage is small even if $T' \gg 2^n$. At the very least, this implies that key-length k of the block cipher E satisfies $k > \log q$. (This is not sufficient of course!) Also we remind here that D' is the description size.

We stress here that we are not focusing on optimizing parameters – and there is a lot of potential for this, by using either better extractors (with shorter seeds) and better sub-key prediction bounds.

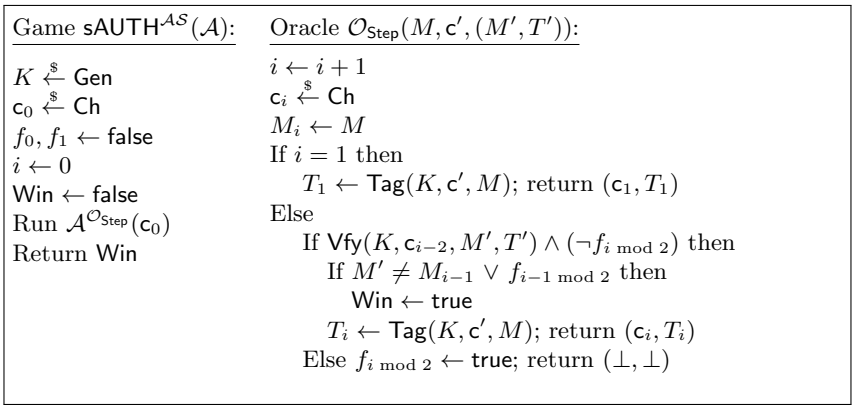

Fig. 3: **Security game sAUTH.** Game defining the security of two-party synchronized authentication. The oracle $\mathcal{O}_{\mathsf{Step}}$ corresponds to each party authenticating chosen messages, in an alternating fashion. Each party will stop answering subsequent queries as soon as a verification query fails. The adversary wins if it delivers a message to a party with a valid tag which was not authenticated by the other party immediately before.

4 Message Authentication

4.1 Synchronous Authentication: Definitions and Settings

We consider the interactive setting of message authentication. Here, two parties alternate communication through an insecure channel (under control of a man-in-the-middle adversary), and want to send authenticated messages to each other. We consider protocols that are *synchronous*, in the sense that at each round one party asks for a challenge c, and the next message M it receives from the other party is authenticated with a tag which depends on *both* c and M (in addition to the secret key). We are not aware of this notion having been extensively studied, but as we will point out below in Section 4.4, considering this setting is somewhat necessary, as building PRFs/MACs secure against memory-bounded adversaries appears out of reach without bypassing existing technical barriers in computational complexity.

Synchronous authentication schemes: Syntax. A *synchronous authentication scheme* is a 4-tuple $\mathcal{AS} = (\mathsf{Gen}, \mathsf{Ch}, \mathsf{Tag}, \mathsf{Vfy})$ of algorithms, which take the following roles:

- The *key generation algorithm* Gen generates a secret key K.
- The *challenge generation algorithm* Ch returns a challenge c.
- The *tagging algorithm* Tag takes as input the secret key K, a message to be authenticated $M \in \mathcal{M}$, and a challenge c, and returns a tag T.
- The *verification algorithm* Vfy takes as input a key K, a challenge c, a message M, and a tag T, and returns a boolean value in $\{\mathsf{true}, \mathsf{false}\}$.

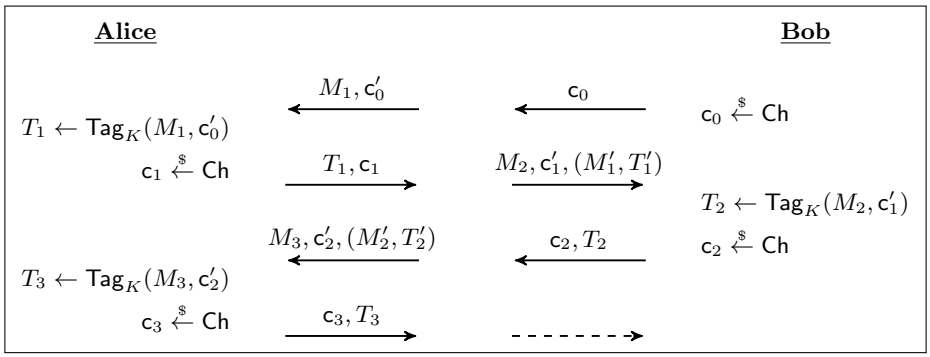

Fig. 4: **Synchronous authentication security game.** This illustrates the flow of the execution of the synchronous authentication game. We omit verification from the figure. At each step, if (M'_i, T'_i) does not verify with respect to c_{i-1}, a pair $(\mathsf{c}_i, T_i) = (\perp, \perp)$ is returned and the corresponding party stops accepting any future messages.

We say that the scheme is ν-correct if for all $M \in \mathcal{M}$,

$$\mathsf{Pr}\left[K \xleftarrow{\$} \mathsf{Gen},\ \mathsf{c} \xleftarrow{\$} \mathsf{Ch},\ T \xleftarrow{\$} \mathsf{Tag}_K(\mathsf{c}, M)\ :\ \mathsf{Vfy}_K(\mathsf{c}, M, T) \neq \mathsf{true}\right] \leq \nu\ .$$

As in the case of encryption, it will be convenient to introduce a notation where we view a function F as the key K. In this case, we write Tag^F and Vfy^F instead of Tag_K and Vfy_K.

Security of authentication schemes. We introduce a security game that captures the security of a synchronous authentication scheme as described above. The game, found in Figure 3, considers an adversary $\mathcal{A}$ interacting with an oracle $\mathcal{O}_{\mathsf{Step}}$, which responds (in an alternating way) as Alice and Bob, each time authenticating a message *chosen* by the adversary. For ease of explanation, a more detailed depiction of the execution flow in the game is given in Figure 4. Then, the advantage of an adversary $\mathcal{A}$ against the authentication scheme $\mathcal{AS}$ is defined as

$$\mathsf{Adv}^{\mathsf{AUTH}}_{\mathcal{AS}}(\mathcal{A}) = \mathsf{Pr}\left[\mathsf{sAUTH}^{\mathcal{AS}}(\mathcal{A}) = \mathsf{true}\right]\ .$$

Further, $\mathsf{Adv}^{\mathsf{AUTH}}_{\mathcal{AS}}(D, T, q, S) = \max_{\mathcal{A}}\{\mathsf{Adv}^{\mathsf{AUTH}}_{\mathcal{AS}}(\mathcal{A})\}$, where the maximum is taken over all S-bounded adversaries $\mathcal{A}$ with running time at most T that makes at most q queries and have size at most D.

As in the case of encryption, in the information-theoretic setting, we drop T and D from the notation and denote the security of the scheme by simply $\mathsf{Adv}^{\mathsf{AUTH}}_{\mathcal{AS}}(q, S) = \mathsf{Adv}^{\mathsf{AUTH}}_{\mathcal{AS}}(\infty, \infty, q, S)$.

4.2 The Challenge-then-Verify Scheme

We give a construction of a synchronous authentication scheme for ℓ-bit messages. The scheme relies on a single function $F : \{0,1\}^n \to \{0,1\}^n$, which we think of being instantiated from a block cipher or a keyed function, but that in the general description we assume comes from a distribution $\mathcal{D}$.

We let t be a parameter, and let $\mathsf{Ext} : \{0,1\}^{t \cdot n} \times \{0,1\}^s \to \{0,1\}^m$ be a function, which should be thought of as an *extractor* later on, and we consequently refer to s as the *seed length*. Also, let $d = \lceil \log(\ell) + 1 \rceil$. We let $\mathcal{I}$ be the uniform distribution over t-tuples of indices $(\mathsf{iv}_1, \ldots, \mathsf{iv}_t) \in \left(\{0,1\}^{n-d-1}\right)^t$. Let $\langle i \rangle$ be the d-bit encoding of $i \in \{1, \ldots, \ell\}$. Generally, we will be interested in the case where $\ell > n$, and s will only depend on n and a desirable security level.

We now describe the algorithms that constitute our authentication scheme *Challenge-then-Verify* $\mathsf{CtV}[\ell, \mathcal{D}, t, \mathsf{Ext}]$. In particular:

Scheme $\mathsf{CtV}[\ell, \mathcal{D}, t, \mathsf{Ext}]$:

- **Key generation.** The key generation algorithm Gen samples F according to distribution $\mathcal{D}$ and outputs F.
- **Challenge generation.** The challenge generation algorithm Ch samples a tuple $(\mathsf{iv}_1, \ldots, \mathsf{iv}_t) \xleftarrow{\$} \mathcal{I}$, as well as a random seed $\mathsf{seed} \xleftarrow{\$} \{0,1\}^s$, and outputs $\mathsf{c} = (\mathsf{iv}_1, \ldots, \mathsf{iv}_t, \mathsf{seed})$.
- **Authentication.** To authenticate a message $M \in \{0,1\}^\ell$ for challenge $\mathsf{c} = (\mathsf{iv}_1, \ldots, \mathsf{iv}_t, \mathsf{seed})$, the tagging algorithm outputs

$$\mathsf{Tag}^F(M = M_1, \ldots, M_\ell, \mathsf{c}) = \bigoplus_{i=1}^{\ell} \mathsf{Ext}(Y_i, \mathsf{seed}) \ ,$$

  where

$$Y_i = F(\langle i \rangle \,\|\, M_i \,\|\, \mathsf{iv}_1) \,\|\, \cdots \,\|\, F(\langle i \rangle \,\|\, M_i \,\|\, \mathsf{iv}_t) \ .$$

- **Verification.** Verification is straightforward, by simply recomputing the tag and checking equality.

When we let $\mathcal{D}$ be the distribution that samples a key K for a block cipher E, and then outputs the function E_K, as above, we denote the resulting scheme simply by $\mathsf{CtV}[\ell, \mathsf{E}, t, \mathsf{Ext}]$.

We will next move to the analysis of the scheme. After that, in Section 4.4, we give some further background about the scheme and possible extensions.

4.3 Security Proof

We first establish the security of the CtV scheme in the information-theoretic setting, where we let the scheme depend on an oracle sampled from a distribution $\mathcal{D}$ on functions from n bits to n bits. To formulate our main theorem, we need to define a derived distribution $\mathcal{D}_{j,b}$ over pairs (F', Z) consisting of a function F'

with corresponding auxiliary information Z. To this end, we sample the function $F : \{0,1\}^n \to \{0,1\}^n$ randomly from $\mathcal{D}$, and then set

$$F' = F_{j,b} \ , \quad Z = \{F_{j',b'}\}_{(j',b') \neq (j,b)}$$

where $F_{j',b'} = F(\langle j' \rangle \,\|\, b' \,\|\, \cdot)$, which is a function $\{0,1\}^{n-d-1} \to \{0,1\}^n$.

This allows us to formulate the following technical theorem. While this is not yet usable to derive bounds with respect to concrete distribution $\mathcal{D}$, as this will require analyzing $\mathcal{D}_{j,b}$, we will give concrete parameter instantiations below.

Theorem 2 (Security of CtV). *For every distribution $\mathcal{D}$ over functions $\{0,1\}^n \to \{0,1\}^n$, if*

$$\max_{j,b} \mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D}_{j,b},t}(S + \ell + m) \leq 2^{-\gamma}$$

and Ext is an average-case (γ, ε)-strong extractor, then

$$\mathsf{Adv}^{\mathsf{AUTH}}_{\mathsf{CtV}[\ell,\mathcal{D},t,\mathsf{Ext}]}(q, S) \leq 4\ell q \left(\frac{1}{2^m} + \varepsilon \right) \ .$$

Proof. Let $\mathcal{A}$ be an S-bounded, q-query adversary for the game $\mathsf{sAUTH}^{\mathsf{CtV}}(\mathcal{A})$, where for simplicity we denote $\mathsf{CtV} = \mathsf{CtV}[\ell, \mathcal{D}, t, \mathsf{Ext}]$. We consider in particular an execution of the S-bounded adversary $\mathcal{A}$, interacting with the oracle $\mathcal{O}_{\mathsf{Step}}$. Following the notation from Figure 4, this interaction defines a sequence of queries consisting of message-challenge pairs

$$(M_1, \mathsf{c}'_0), (M_2, \mathsf{c}'_1), \ldots, (M_q, \mathsf{c}'_{q-1}) \ ,$$

as well as forgery attempts

$$(M'_2, T'_2), \ldots, (M'_q, T'_q) \ .$$

These come with corresponding query answers $(\mathsf{c}_1, T_1), \ldots, (\mathsf{c}_q, T_q)$, where recall that $(\mathsf{c}_i, T_i) = (\bot, \bot)$ if $\mathcal{O}_{\mathsf{Step}}$ fails to return an answer. Further, for any i and j, we denote by $M_{i,j}$ and $M'_{i,j}$, respectively, the j-th bit of M_i and M'_i. Also, we let $\sigma_0, \sigma_1, \ldots, \sigma_q$ be the sequence of states of $\mathcal{A}$ during this execution. We can assume without loss of generality that $\mathcal{A}$ is deterministic, by hard-coding the optimal randomness in the description of $\mathcal{A}$, as our arguments will be independent of the *size* of $\mathcal{A}$. (Thus, the length of the fixed randomness does not count towards the memory resources of $\mathcal{A}$.)

We define the family of events $\mathsf{Win}_{i,j,b,d}$ where $i \in [q] \setminus \{1\}$, $j \in [\ell]$, $d, b \in \{0,1\}$. Here, $\mathsf{Win}_{i,j,b,d}$ is the event that the following conditions are simultaneously true:

(1) The adversary $\mathcal{A}$ provokes $\mathsf{Win} \leftarrow \mathsf{true}$ in the i-th query (and thus Win was false up to that point);
(2) $b = M'_{i,j}$
(3) If $d = 1$, the $(i-1)$-th query did not return $(\bot, \bot)$. Further, $M_{i-1,j} = 1 - b$, and $M_{i-1,j'} = M'_{i,j'}$ for all $j' < j$. That is M'_i and M_{i-1} differ in the j-th bit, which takes value b and $1 - b$ respectively, and M'_i and M_{i-1} are identical on the first $j - 1$ bits.

(4) If $d = 0$, the $(i-1)$-th query returned $(\perp, \perp)$.

Then, we clearly have[8]

$$\mathsf{Adv}^{\mathsf{AUTH}}_{\mathsf{CtV}}(\mathcal{A}) = \sum_{i=2}^{q} \sum_{j=1}^{\ell} \sum_{b,d \in \{0,1\}} \mathsf{Pr}\left[\mathsf{Win}_{i,j,b,d}\right] \ . \tag{3}$$

We are going to now upper bound each individual probability $\mathsf{Pr}\left[\mathsf{Win}_{i,j,b,d}\right]$ in terms of the sub-key prediction advantage.

Reduction to sub-key prediction. Fix i, j, b, d. We first consider a variant of the sub-key prediction game where the goal is to predict the value of Ext applied to the sub-key, rather than predicting the sub-key itself. The game involves an adversary $\mathcal{B}$ and a leakage function $\mathcal{L}$, which we specify below, and the distribution $\mathcal{D}_{j,b}$ is as defined above:

- $(F_{j,b}, Z) \xleftarrow{\$} \mathcal{D}_{j,b}$
- $\sigma \leftarrow \mathcal{L}(F_{j,b}, Z)$
- $(\mathsf{iv}_1, \ldots, \mathsf{iv}_t) \xleftarrow{\$} \mathcal{I}$
- $\mathsf{seed} \xleftarrow{\$} \{0,1\}^s$
- $T \leftarrow \mathcal{B}(\sigma, Z, i_1, \ldots, i_t, \mathsf{seed})$
- Return $(T = \mathsf{Ext}(F_{j,b}(\mathsf{iv}_1) \| \ldots \| F_{j,b}(\mathsf{iv}_t), \mathsf{seed}))$

We stress that the game returns true if and only if T equals the extractor output. It is convenient to denote by $p_{\mathcal{B},\mathcal{L}}$ the probability that this is indeed the case. We now give $\mathcal{B}$ and $\mathcal{L}$ such that

$$\mathsf{Pr}\left[\mathsf{Win}_{i,j,b,d}\right] \leq p_{\mathcal{B},\mathcal{L}} \ . \tag{4}$$

Concretely, leakage function $\mathcal{L}$ is given access to the description of 2ℓ functions $F_{1,1}, F_{0,1}, \ldots, F_{\ell,0}, F_{\ell,1}$ through $(F_{j,b}, Z = \{F'_{j',b'}\}_{(j',b') \neq (j,b)})$. It simulates correctly the execution of $\mathcal{A}$ in Game $\mathsf{sAUTH}^{\mathsf{CtV}}(\mathcal{A})$ for the first $i-2$ queries to $\mathcal{O}_{\mathsf{Step}}$, using the 2ℓ functions. The $(i-2)$-th query returns in particular a tag T_{i-2} for the message M_{i-2} and challenge c'_{i-3} – here we ignore the associated challenge c_{i-2} (with some foresight, we will simulate it from $\mathcal{B}$'s input) – and note that $T_{i-2} = \perp$ is possible. The leakage function then outputs $(\sigma_{i-2}, M_{i-2}, T_{i-2})$, where σ_{i-2} is $\mathcal{A}$'s state when making the $(i-2)$-th query.

Then, the adversary $\mathcal{B}$ is now given the leakage $(\sigma_{i-2}, M_{i-2}, T_{i-2})$, the auxiliary information $Z = \{F'_{j',b'}\}_{(j',b') \neq (j,b)}$, as well as a fresh $(\mathsf{iv}_1, \ldots, \mathsf{iv}_t)$ and seed. The only thing $\mathcal{B}$ does not know is $F_{j,b}$. Then, $\mathcal{B}$ proceeds through the following steps:

1. $\mathcal{B}$ resumes the execution of $\mathcal{A}$ with input σ_{i-2}, M_{i-2}, T_{i-2}, and $\mathsf{c}_{i-2} = (i_1, \ldots, i_t, \mathsf{seed})$ (if $T_{i-2} \neq \perp$) or $\mathsf{c}_{i-2} = \perp$ (if $T_{i-2} = \perp$).
2. When $\mathcal{A}$ asks the $(i-1)$-th query to $\mathcal{O}_{\mathsf{Step}}$ with the format $(M_{i-1}, \mathsf{c}'_{i-2}, (M'_{i-1}, T'_{i-1}))$, we distinguish between two cases.

[8] Note that the fact that we have equality is not really important here, but the events indeed happen to be disjoint.

(a) First, if $d = 0$, $\mathcal{B}$ returns $(\bot, \bot)$ to the simulated $\mathcal{A}$.
(b) If $d = 1$, $\mathcal{B}$ stops outputting a random m-bit guess if $M_{i-1,j} \neq 1 - b$. Otherwise, it computes $T_{i-1} \leftarrow \mathsf{Tag}^F(M_{i-1}, \mathsf{c}'_{i-2})$. Note that because $M_{i-1,j} = 1 - b$, this can be done with the available functions within Z, since $F_{j,b}$ is not involved in the computation. It then returns $(T_{i-1}, \mathsf{c}_{i-1})$ to $\mathcal{A}$.

3. Finally, $\mathcal{A}$ outputs its i-th query $(M_i, \mathsf{c}'_{i-1}, (M'_i, T'_i))$. Now, if $M'_{i,j} \neq b$, $\mathcal{B}$ stops with a random m-bit guess. Otherwise, we compute, for all $j' \neq j$,

$$Y_{j'} = F_{j',M'_{i,j'}}(\mathsf{iv}_1) \,\|\, \cdots \,\|\, F_{j',M'_{i,j'}}(\mathsf{iv}_t) \;,$$

and finally output the guess

$$T = T'_i \oplus \bigoplus_{j' \neq j} \mathsf{Ext}(Y_{j'}, \mathsf{seed}) \;.$$

It now clear that by construction Equation 4 is always satisfied. This is because provided $\mathsf{Win}_{i,j,b,d}$ occurs, we can map an execution from $\mathsf{sAUTH}^{\mathsf{CtV}}(\mathcal{A})$ into one where $\mathcal{L}$ and $\mathcal{B}$ correctly guess Ext's output.

To conclude the proof, we note that $\mathcal{L}$'s output has length $S + \ell + m$ bits, and therefore, because $\mathsf{Adv}^{\mathsf{skp\text{-}aux}}_{\mathcal{D}_{j,b},t}(S + \ell + m) \leq 2^{-\gamma}$, by Lemma 4,

$$\mathrm{H}_\infty\left(F_{j,b}(\mathsf{iv}_1) \| \ldots \| F_{j,b}(\mathsf{iv}_t) | \sigma_{i-2}, (\mathsf{iv}_1, \ldots \mathsf{iv}_t)\right) \geq \gamma \;.$$

But because Ext is a (γ, ε)-strong extractor, this also implies that

$$(\mathsf{Ext}(F_{j,b}(\mathsf{iv}_1) \| \ldots \| F_{j,b}(\mathsf{iv}_t), \mathsf{seed}), \sigma_{i-2}, (\mathsf{iv}_1, \ldots \mathsf{iv}_t), \mathsf{seed})$$

and

$$(Z, \sigma_{i-2}, (\mathsf{iv}_1, \ldots \mathsf{iv}_t), \mathsf{seed})$$

for uniformly distributed $Z \xleftarrow{\$} \{0,1\}^m$, have statistical distance at most ε. Therefore,

$$\mathsf{Pr}\left[\mathsf{Win}_{i,j,b,d}\right] \leq p_{\mathcal{B},\mathcal{L}} \leq \varepsilon + \frac{1}{2^m} \;.$$

This also concludes the proof, by plugging this into Equation 3. □

Instantiations. With the goal of providing a block-cipher based instantiation of the construction, we consider the case where $\mathcal{D}$ is the uniform distribution over all n-bit permutations. Then, note that $F_{j,b}$, given $F_{j',b'}$ for (j', b'), is still uniformly distributed over a set of $2^{n-d-1}!$ possible functions.

Corollary 2. *Let* $\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ *be a block cipher. Let* $\mathcal{H} : \{0,1\}^{tn} \times \{0,1\}^{tn} \to \{0,1\}^m$ *be a 2-universal family of hash functions. Let* $S + \ell + m \leq N + \frac{N(n - \log(16\ell))}{8\ell} - \delta n N$ *for some* $\delta \in [0,1]$.

Then, if $\delta \geq \left[\frac{4(m - 2\log\varepsilon)}{t} + \frac{1}{n} + \frac{3(m - 2\log\varepsilon) + 5}{nN}\right]$ *for some* $\varepsilon > 0$, *then for all* D, T, *there exists* $D' \approx D$ *and* $T' \approx T$ *such that*

$$\mathsf{Adv}^{\mathsf{AUTH}}_{\mathsf{CtV}[\ell,\mathcal{I},t,\mathcal{H},\mathsf{E}]}(D,T,q,S) \leq 4\ell q\left(\frac{1}{2^m}+\varepsilon\right)+\mathsf{Adv}^{\mathsf{PRP-CPA}}_{\mathsf{E}}(D',T',t\ell q,S').$$

where $S' = S + 2tn + 2\ell + m$.

Beyond 2^n-security. Again, to demonstrate that our authentication scheme can tolerate queries beyond $q = 2^n$ by the adversary and still have meaningful security, we plug in concrete values in Corollary 2. Let $q \leq 2^{1.5n}$ and $\ell = 2n$. Let the output of the extractor be of length $m = 3n$. Say we want ε to be 2^{-3n} such that $4\ell q\left(\frac{1}{2^m}+\varepsilon\right) \leq 8n2^{-1.5n}$ when an S-bounded adversary is such that $S \leq N^{2/3}$. Then, by plugging in the desired parameters, we can see that for $n \geq 10$, we achieve the preferred security bound at $t \geq 300n^2$.

4.4 Remarks and Extensions

We give here a few remarks about our construction above. We will first discuss why a stronger result (dispensing with challenges) appears hard. We then discuss briefly how to extend the domain of authenticated messages, and the combination of encryption and authentication.

Building PRFs: Why is it hard? An excellent question is whether we can build an actual PRF (and consequently a MAC), thus dispensing with the need for a challenge. The natural approach is to extend the domain of a random function[9] $R : \{0,1\}^n \to \{0,1\}^n$ to a function $\mathsf{F}^R : \{0,1\}^m \to \{0,1\}^n$ where $m > n$, which is indistinguishable from a truly random function for $q \gg 2^n$ queries, provided the distinguisher's memory is bounded by $S < 2^n$. This appears well beyond reach of current techniques, and would require overcoming barriers in the design of PRGs against space-bounded computation.

Specifically, consider a function $G : \{0,1\}^k \to \{0,1\}^\ell$ where $k > \ell$, and we now look at a model where, for a random $x \xleftarrow{\$} \{0,1\}^k$, a distinguisher is given oracle access to either the ℓ individual bits $y_1 \dots y_\ell = G(x)$ or to independent random bits $y_1, \dots, y_\ell$. The function G is an ε-PRG for S-bounded distinguishers if every space-S distinguisher can only succeed in distinguishing the two cases with advantage ε. Clearly, $S < k$ must hold, and the state of the art constructions [16,27] achieve $\ell = O(k)$, even if we only demand $\varepsilon = 1/\omega(\log(k))$.[10]

A domain extender F described above would in particular define an ε-PRG $G = G^{\mathsf{F}}$ for S-bounded computation with $k = n \cdot 2^n$ and $\ell = q \cdot n$ and $\varepsilon = n^{-\omega(1)}$. The PRG would just interpret its seed x as a function $f : \{0,1\}^n \to \{0,1\}^n$, and output a sequence of bits obtained by evaluating F^f at q distinct inputs. If $q \geq 2^{n(1+\delta)}$ for a constant $\delta > 0$, then we have $\ell \approx k^{1+\delta}$. Also, because F can only make a small number $t = \mathsf{poly}(n)$ of calls to f, the resulting PRG G is

[9] Or a permutation, but we restrict ourselves to functions as this only makes the problem easier, and our point stronger.

[10] We note that much better constructions exist if one imposes restrictions on the distinguisher's queries, e.g., the bits are read once from y_1 to y_ℓ.

local, in the sense that every output bit only depends on $O(\log(k))$ bits of the seed. Existing constructions [16,27] have only linear stretch and are inherently non-local.

HIGHER EFFICIENCY. There is nothing really special about the scheme processing the message one bit at a time. The analysis can easily be generalized so that the scheme processes a large number of bits per call. That is, we would have for each $i \in [\ell]$, where now ℓ is the number of b-bit blocks, and the i-th block M_i,

$$Y_i = F(\langle i \rangle \,\|\, M_i \,\|\, \mathsf{iv}_1) \,\|\, \cdots \,\|\, F(\langle i \rangle \,\|\, M_i \,\|\, \mathsf{iv}_t) \ .$$

We would lose in security, as the iv-values are now shorter, i.e., $n - b - d$, but this gives acceptable compromises. The analysis is a straightforward adaptation of the one we have given above.

EXTENDING THE DOMAIN. Our scheme above authenticates messages of fixed length ℓ. It can however straightforwardly be extended to authenticate arbitrarily long messages if we assume a collision resistant hash function family producing ℓ-bit hashes, for a sufficiently long ℓ, which is more secure than the underlying PRP E. For example, if the key length is k bits, one could assume $\ell = 2k$ and that collisions can only be found in time 2^k.

AUTHENTICATED ENCRYPTION. We will not discuss this in detail here, but clearly encryption and authentication can be combined to obtain a resulting notion of (synchronous) authenticated encryption. The messages to be authenticated would be ciphertexts produced with the encryption scheme from Section 3, and both schemes would use two independent keys.

5 Key-Length Extension in the Memory-Bounded Setting

5.1 Problem Formulation

The results from the previous sections require a block cipher with security beyond 2^n queries. This in particular requires a long key, and we may not have it (e.g., in AES-128, the key length equals the block length). The classical problem of key-length extension addresses exactly this – several solutions have been validated in the ideal-cipher model [15,21,22,23,24,26,28],[11] and are commonly assumed to work with a good block cipher. Such results however assume no bounds on the adversary's memory, and thus, if we assume the adversary can store fewer than 2^n bits, they may be overly pessimistic. To this end, here, we analyze the security of double encryption in the ideal cipher model when the memory of the adversary is bounded. Double encryption is particularly interesting, because it is known *not* to amplify security when the memory of the adversary is unbounded. We will see that when the memory of the attacker does not exceed 2^k, for a k-bit key, things are substantially different, at least under reasonable assumptions.

[11] We note that the use of the ideal-cipher model is somehow necessary, as we are achieving effectively true hardness amplification.

DEFINITIONS. Let $\mathsf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Then, the double encryption scheme $\mathsf{DE} = \mathsf{DE}[\mathsf{E}]$ is the block cipher such that

$$\mathsf{DE}_{K_1,K_2}(x) = \mathsf{E}_{K_2}(\mathsf{E}_{K_1}(x)) \tag{5}$$

where $K_1, K_2 \in \{0,1\}^k$. Clearly, $\mathsf{DE}^{-1}_{K_1,K_2}(y) = \mathsf{E}^{-1}_{K_1}(\mathsf{E}^{-1}_{K_2}(y))$.

The security notion considered for the double encryption scheme is that of *strong* PRP-security, where the attacker can make both forward and backwards queries. We will consider it in particular in the ideal-cipher model – to this end, let $\mathcal{BC}_{k,n}$ be the set of all block ciphers with key length k and block length n. The adversary has access to two pairs of oracles:

1. An ideal cipher oracle $\mathsf{E} \xleftarrow{\$} \mathcal{BC}_{k,n}$ and its inverse E^{-1} s.t. $\mathsf{E}^{-1}(K', y) = \mathsf{E}^{-1}_{K'}(y)$.
2. An oracle $\mathcal{O}$ and its inverse $\mathcal{O}^{-1}$, where $\mathcal{O}/\mathcal{O}^{-1} : \{0,1\}^n \to \{0,1\}^n$. The oracle $\mathcal{O}$ is either the double encryption scheme $\mathsf{DE}_{K_1,K_2}(\cdot) = \mathsf{E}_{K_2}(\mathsf{E}_{K_1}(\cdot))$ with uniform, independent, keys K_1 and K_2 (in the real world) or a random permutation $P \xleftarrow{\$} \mathcal{P}_n$ (in the ideal world).

At the end of q steps, the adversary tries to guess if the oracle $\mathcal{O}$ it has been interacting with is DE_{K_1,K_2} or P.

More explicitly, the advantage of an adversary $\mathcal{A}$ against the double encryption scheme $\mathsf{DE}[\mathsf{E}]$ is defined as

$$\begin{aligned}\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{DE}[\mathsf{E}]}(\mathcal{A}) = \big|\mathsf{Pr}[K_1, K_2 \xleftarrow{\$} \{0,1\}^k, \mathsf{E} \xleftarrow{\$} \mathcal{BC}_{k,n} \ : \ \mathcal{A}^{\mathsf{DE}_{K_1,K_2},\mathsf{DE}^{-1}_{K_1,K_2},\mathsf{E},\mathsf{E}^{-1}} = 1] \\ - \mathsf{Pr}[P \xleftarrow{\$} \mathcal{P}_n \ : \ \mathcal{A}^{P,P^{-1},\mathsf{E},\mathsf{E}^{-1}} = 1]\big| \ .\end{aligned}$$

5.2 Double Encryption and List Disjointness

We study the security of the double encryption scheme in our model by relating it to a problem that we introduce, called the *list disjointness* problem – this is a special case of the element distinctness problem studied in the literature. We show that any algorithm solving this problem immediately implies an attacker against double encryption (with the same complexity). More importantly, as our main result, we show that any attacker against double encryption also implies an algorithm solving list disjointness.

THE LIST DISJOINTNESS PROBLEM. The setting for the $\mathrm{LD}_{\kappa,k}$ problem is as follows: An algorithm is given oracle access to two lists L_1 and L_2, each containing $\kappa/2$ distinct k-bit elements, i.e., the algorithm can learn the j-th element of L_i by making a query $L_i[j]$ for $i \in \{1,2\}$ and $j \in \{1,\ldots,\ell\}$. The lists are such that they have at most one element in common, i.e., we have the promise that $|L_1 \cap L_2| = 1$ or $|L_1 \cap L_2| = 0$. The aim of the algorithm is to distinguish the two cases given oracle access to the two lists. The list disjointness problem is a special case of the element distinctness problem where given oracle access to a list, an algorithm tries to determine whether all elements in the list are distinct. The following definition formalizes this as a distinguishing problem.

Definition 2 (List Disjointness Problem). *An algorithm* Alg *with binary output is said to solve the* list disjointness *problem* $LD_{\kappa,k}$ *with advantage* ε *if it is given oracle access to two lists* L_1, L_2 *of* $\kappa/2$ k*-bit elements (which we can think of as functions* $L_1, L_2 : [\kappa/2] \to \{0,1\}^k$*) such that* $|L_1 \cap L_2| \leq 1$*, and, moreover, for any such* L_1, L_2*, the difference between the probabilities that* Alg *outputs* 1 *when* $|L_1 \cap L_2| = 1$ *and when* $L_1 \cap L_2 = \emptyset$ *is* at least ε*.*

We note that advantage above can be amplified via sequential repetition – this requires minimal memory overhead to estimate the number of repetitions outputting one. We omit the details.

List Disjointness to DE. We first observe that an algorithm Alg that solves the list disjointness problem immediately implies a distinguisher against the PRP-security of the double encryption scheme with similar memory and time complexities, and advantage. This can be seen as follows. The distinguisher runs Alg and provides oracle access to two lists L_1 and L_2 where the lists are each of size 2^k, and each index j in L_i is associated with a unique k-bit string $K^j \in \{0,1\}^k$. The distinguisher makes a constant c number of queries to its permutation oracle (that is either DE_{K_1,K_2} or P) to obtain plaintext/ciphertext pairs $(x_1, y_1), \ldots, (x_c, y_c)$. (The constant c is related to the ratio between key length and block length of the block cipher E.) Now, when Alg queries the list L_i at index j, the distinguisher answers this query using its $\mathsf{E}/\mathsf{E}^{-1}$ oracle as follows:

- if $i = 1$, return $\mathsf{E}_{K^j}(x_1) \| \ldots \| \mathsf{E}_{K^j}(x_c)$ as the element $L_1[j]$ and
- if $i = 2$, return $\mathsf{E}^{-1}_{K^j}(y_1) \| \ldots \| \mathsf{E}^{-1}_{K^j}(y_c)$ as the element $L_2[j]$.

When the permutation oracle of the distinguisher is the double encryption oracle, L_1 and L_2 share exactly an element, while if it were a random permutation, an element is shared only with probability negligible in k.

DE to List Disjointness. The reduction for transforming an adversary against the double encryption scheme to an algorithm for list disjointness is more involved. In fact, our algorithm in the list disjointness problem will require access to additional oracles that can be queried for free (i.e., such queries do not count towards the query complexity). Specifically, it will use:

- A permutation $\rho : [\kappa] \to [\kappa]$ chosen uniformly from the set of all permutations over $[\kappa]$. On input K, the output $\rho(K)$ is interpreted as $\rho(K) = (i, j)$ where $i \in \{1, 2\}$ and $j \in [\kappa/2]$.
- A permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ chosen uniformly from the set of all permutations over $\{0,1\}^n$, and its inverse π^{-1}.
- An ideal cipher $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$.

We stress that these oracles do not depend on the lists L_1 and L_2. (In a heuristic implementation they could be realized e.g., from a block cipher.)

Given an adversary $\mathcal{A}$ against double encryption achieving advantage ε, we show how to solve the list disjointness problem with advantage $\varepsilon - 2^k$, given access to F, ρ, and π as defined above.

Theorem 3. *Let $\mathcal{A}$ be an S-bounded attacker making at most q ideal-cipher queries (and any number of queries to its $\mathcal{O}$ / $\mathcal{O}^{-1}$ oracle) such that*

$$\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{DE}[\mathsf{E}]}(\mathcal{A}) \geq \varepsilon \ ,$$

where the underlying ideal cipher has key length k and block length n. Then, there exists an S-bounded algorithm Alg *that makes q queries to the given lists, uses the oracles ρ, F, π defined above, and solves the list disjointness problem $LD_{\kappa=2^k,k}$ with advantage $\varepsilon - 2^{-k}$.*

Proof (Sketch). Fix an adversary $\mathcal{A}$ against the double encryption scheme DE such that it has the maximum advantage. We assume without loss of generality that the probability it outputs 1 in the real world is at least ε higher than in the ideal world. Recall that the algorithm Alg has access to oracles L_1, L_2, ρ, π, F as mentioned in Definition 2. The algorithm proceeds by running $\mathcal{A}$, and thus it is required to simulate the ideal cipher and permutation oracles that $\mathcal{A}$ expects access to. This is done in the following manner. If $\mathcal{A}$ queries the permutation oracle $\mathcal{O}$ or $\mathcal{O}^{-1}$, the algorithm Alg just returns the answer by querying its random permutation oracle π or its inverse π^{-1}. A query $(K, \cdot)$ to the ideal cipher oracle on key K is answered as follows: We interpret $\rho(K)$ as (i, j) where $i \in \{1, 2\}$ and $j \in [\kappa/2]$. Then, if $i = 1$:

- a forward query (K, x) is answered as $\mathsf{E}'_K(x) \leftarrow F_{L_1[j]}(x)$,
- an inverse query (K, y) is answered as $\mathsf{E}'^{-1}_K(y) \leftarrow F^{-1}_{L_1[j]}(y)$.

If $i = 2$:

- a forward query (K, x) is answered as $\mathsf{E}'_K(x) \leftarrow \pi(F^{-1}_{L_2[j]}(x))$,
- an inverse query (K, y) is answered as $\mathsf{E}'^{-1}_K(y) \leftarrow F_{L_2[j]}(\pi^{-1}(y))$.

At the end, Alg outputs $\mathcal{A}$'s output bit.

We now note the following, omitting a formal argument:

- If the lists L_1 and L_2 do not intersect, then the keys on which F is called for the cases $i = 1$ and $i = 2$ are distinct, and thus we are perfectly simulating the ideal world, since composing π with F in the $i = 2$ case does not change the distribution of the query answers.
- If the lists L_1 and L_2 intersect exactly at one point, then there are two distinct keys K_1 and K_2 such that $\rho(K_1) = (1, j_1)$, $\rho(K_2) = (2, j_2)$, and $L_1[j_1] = L_2[j_2]$. This ensures that $\mathsf{E}_{K_1}(\mathsf{E}_{K_2}(x)) = \pi(x)$. Moreover, because ρ is a random permutation, K_1 and K_2 are uniformly distributed, conditioned on $K_1 \neq K_2$. Thus, we are simulating the real world *conditioned* on $K_1 \neq K_2$.

Therefore, as claimed, Alg solves the list disjointness problem with advantage at least $\varepsilon - 2^{-k}$. □

State-of-the-art for List Disjointness. Now that we have shown that an attacker against double encryption leads to an algorithm solving list disjointness with similar complexity, we state the best existing algorithm for list disjointness and conjecture that this is the best possible.

To this end, we first state the following result by Beame, Clifford, and Machmouchi [8] that gives an algorithm for computing element distinctness. In the following statement ED_n refers to the decision problem where given n elements belonging to some domain we need to determine if the n elements are distinct or not. Again, the advantage will measure the difference between the probability of a positive answer when the elements are distinct and when they are not. As a corollary of this result, we can derive a time-space upper bound for the list disjointness problem mentioned above.

Theorem 4 ([8]). *For any $\varepsilon > 0$, and any S with $c \log n \leq S \leq n/32$ for some constant $c > 0$, there is an S-bounded algorithm solving ED_n with advantage ε making $q = O\left(\frac{n^{3/2}}{S^{1/2}} \log^{5/2} n \log(1/(1-\varepsilon))\right)$ queries to the given list.*

This theorem immediately gives us the following corollary as list disjointness can be seen as a special case of the element distinctness problem where the elements under consideration are those belonging to the two lists.

Corollary 3. *For any $\varepsilon > 0$, and any S with $c \log \kappa \leq S \leq \kappa/32$ for some constant $c > 0$, there is an S-bounded algorithm solving $LD_{\kappa,k}$ with advantage ε, and making*

$$q = O\left(\frac{\kappa^{3/2}}{S^{1/2}} \log^{5/2}(\kappa) \log(1/(1-\varepsilon))\right)$$

queries.

We have been somewhat informal here, as the algorithm of [8] actually requires access to a random hash function. This can be implemented from the oracles made available in our extended setting of $LD_{\kappa,k}$.

We note that finding good lower bounds for the element distinctness problem has been a major open problem in complexity theory for the past three decades and progress has been slow on that front. The best known lower bound is due to Beame et al. [10] that showed $T \in \Omega\left(n\sqrt{\log(n/S)/\log\log(n/S)}\right)$. A better lower bound of $T \in \Omega\left(n^{2-o(1)}/S\right)$ was given by Yao [39] in the restricted setting of *comparison branching programs* (where access to the input is limited to pairwise comparison). Until the result stated in Theorem 4, it was not known whether the lower bound in the general setting matches the restricted setting given by Yao [39].

A conditional lower bound. Given the current state-of-the-art, we conjecture that the result by Beame et al. [8] does in fact provide the best algorithm for computing element distinctness and hence assume that it gives a lower bound on the time-space tradeoff for the element distinctness problem. We state that following (slightly more conservative) conjecture (note that we have implicitly used that $\log(1/(1-\varepsilon)) = \Omega(\varepsilon)$ here).

Conjecture 1. There are constants c_1, c_2, such that for any $\varepsilon > 0$ and any S with $c_1 \log \kappa \leq S \leq \kappa/c_2$, every S-bounded algorithm to solve the list disjointness problem $LD_{\kappa,k}$ with advantage at least ε requires querying the lists

$$q = \Omega\left(\frac{\kappa^{3/2}}{S^{1/2}}\varepsilon\right)$$

times.

Therefore, under Conjecture 1, Theorem 3 directly yields a lower bound, and in particular for any S-bounded attacker $\mathcal{A}$ that queries the ideal cipher at most $q = O\left(\frac{2^{3k/2}}{S^{1/2}}(\varepsilon - 2^{-k})\right)$ times, the advantage is at most ε, or equivalently, for any S-bounded $\mathcal{A}$ making at most q queries to the ideal cipher,

$$\mathsf{Adv}^{\mathsf{PRP}}_{\mathsf{DE}[\mathsf{E}]}(\mathcal{A}) = O\left(\sqrt{\frac{Sq^2}{2^{3k}}}\right) + \frac{1}{2^k}\ .$$

We stress that the bound is independent of the number of queries to the $\mathcal{O}$ / $\mathcal{O}^{-1}$ oracle. Note that if $S = 2^k$, we recover the traditional bound of $q/2^k$, which is tight by the meet-in-the-middle attack. (It is worth noting that Aiello et al. [1] show the slightly superior bound of $(q/2^k)^2$ here.) However, if for example $S = 2^{k/2}$, then we get security up to $q = 2^{1.25k}$ queries.

Acknowledgments. Stefano Tessaro's work was partially supported by NSF grants CNS-1553758 (CAREER), CNS-1423566, CNS-1719146, CNS-1528178, and IIS-1528041, and by a Sloan Research Fellowship. Aishwarya Thiruvengadam's work was partially supported by the Defense Advanced Research Projects Agency (DARPA) and Army Research Office (ARO) under Contract No. W911NF-15-C-0236, and a subcontract No. 2017-002 through Galois.

References

1. William Aiello, Mihir Bellare, Giovanni Di Crescenzo, and Ramarathnam Venkatesan. Security amplification by composition: The case of doubly-iterated, ideal ciphers. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 390–407, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Heidelberg, Germany.
2. Miklós Ajtai. A non-linear time lower bound for boolean branching programs. *Theory of Computing*, 1(8):149–176, 2005.
3. Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Depth-robust graphs and their cumulative memory complexity. In Coron and Nielsen [18], pages 3–32.
4. Joël Alwen, Binyi Chen, Krzysztof Pietrzak, Leonid Reyzin, and Stefano Tessaro. Scrypt is maximally memory-hard. In Coron and Nielsen [18], pages 33–62.
5. Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 36–54, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Heidelberg, Germany.

6. Joël Alwen and Vladimir Serbinenko. High parallel complexity graphs and memory-hard functions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *47th ACM STOC*, pages 595–603, Portland, OR, USA, June 14–17, 2015. ACM Press.
7. Benedikt Auerbach, David Cash, Manuel Fersch, and Eike Kiltz. Memory-tight reductions. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 101–132, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.
8. Paul Beame, Raphaël Clifford, and Widad Machmouchi. Element distinctness, frequency moments, and sliding windows. In *54th FOCS*, pages 290–299, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press.
9. Paul Beame, Michael Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *J. ACM*, 50(2):154–195, March 2003.
10. Paul Beame, Michael E. Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *J. ACM*, 50(2):154–195, 2003.
11. Mihir Bellare and Wei Dai. Defending against key exfiltration: Efficiency improvements for big-key cryptography via large-alphabet subkey prediction. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 17*, pages 923–940, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press.
12. Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.
13. Mihir Bellare, Oded Goldreich, and Hugo Krawczyk. Stateless evaluation of pseudorandom functions: Security beyond the birthday barrier. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 270–287, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Heidelberg, Germany.
14. Mihir Bellare, Daniel Kane, and Phillip Rogaway. Big-key symmetric encryption: Resisting key exfiltration. In Robshaw and Katz [34], pages 373–402.
15. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany.
16. Andrej Bogdanov, Periklis A. Papakonstantinou, and Andrew Wan. Pseudorandomness for linear length branching programs and stack machines. In *APPROX-RANDOM*, volume 7408 of *Lecture Notes in Computer Science*, pages 447–458. Springer, 2012.
17. Allan Borodin, Michael J. Fischer, David G. Kirkpatrick, Nancy A. Lynch, and Martin Tompa. A time-space tradeoff for sorting on non-oblivious machines. *Journal of Computer and System Sciences*, 22(3):351 – 364, 1981.
18. Jean-Sébastien Coron and Jesper Buus Nielsen, editors. *EUROCRYPT 2017, Part II*, volume 10211 of *LNCS*, Paris, France, May 8–12, 2017. Springer, Heidelberg, Germany.
19. Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.
20. Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space lower bounds for learning. *CoRR*, abs/1708.02639, 2017.

21. Peter Gaži. Plain versus randomized cascading-based key-length extension for block ciphers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 551–570, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany.
22. Peter Gazi, Jooyoung Lee, Yannick Seurin, John P. Steinberger, and Stefano Tessaro. Relaxing full-codebook security: A refined analysis of key-length extension schemes. In Gregor Leander, editor, *FSE 2015*, volume 9054 of *LNCS*, pages 319–341, Istanbul, Turkey, March 8–11, 2015. Springer, Heidelberg, Germany.
23. Peter Gaži and Ueli M. Maurer. Cascade encryption revisited. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 37–51, Tokyo, Japan, December 6–10, 2009. Springer, Heidelberg, Germany.
24. Peter Gaži and Stefano Tessaro. Efficient and optimally secure key-length extension for block ciphers via randomized cascading. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 63–80, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.
25. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.
26. Viet Tung Hoang and Stefano Tessaro. Key-alternating ciphers and key-length extension: Exact bounds and multi-user security. In Robshaw and Katz [34], pages 3–32.
27. Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. In *53rd FOCS*, pages 111–119, New Brunswick, NJ, USA, October 20–23, 2012. IEEE Computer Society Press.
28. Jooyoung Lee. Towards key-length extension with optimal security: Cascade encryption and xor-cascade encryption. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 405–425, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.
29. Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.
30. Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43 – 52, 1996.
31. J. M. Pollard. A monte carlo method for factorization. *BIT Numerical Mathematics*, 15(3):331–334, Sep 1975.
32. Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. In Irit Dinur, editor, *57th FOCS*, pages 266–275, New Brunswick, NJ, USA, October 9–11, 2016. IEEE Computer Society Press.
33. Ran Raz. A time-space lower bound for a large class of learning problems. In *58th FOCS*, pages 732–742. IEEE Computer Society Press, 2017.
34. Matthew Robshaw and Jonathan Katz, editors. *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany.
35. John E. Savage. *Models of Computation: Exploring the Power of Computing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1997.
36. Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, January 2004.
37. Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012.
38. Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12(1):1–28, 1999.

39. Andrew Chi-Chih Yao. Near-optimal time-space tradeoff for element distinctness. In *29th FOCS*, pages 91–97, White Plains, New York, October 24–26, 1988. IEEE Computer Society Press.
40. Andrew Chi-Chih Yao. Near-optimal time-space tradeoff for element distinctness. *SIAM Journal on Computing*, 23(5):966–975, 1994.