Limits of Pseudorandom Codes under Adaptive Corruption

Ganyuan Cao[1]

Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
ganyuan.cao@telecom-paris.fr

Abstract. Pseudorandom codes (PRCs) hide messages in strings that look uniformly random to anyone without the decoder key. If the key may be stolen after some codewords have been sent, security requires that these codewords and the stolen key be indistinguishable from uniform strings, chosen independently of the messages, together with a key that decodes each of them to its message. We call producing such a key *opening* the codewords, and ask how many key bits it takes.

We first show that every adaptively secure PRC yields a *q-use code* with the same key length, namely its opened key together with the q uniform strings it opens. Lower bounds for these information-theoretic objects therefore hold unconditionally and apply to every adaptively secure PRC, including current designs based on hardness assumptions.

We then show that every opening costs key bits beyond the opened message: $\omega(\log \lambda)$ without errors, $\frac{1}{\rho^2} \cdot \omega(\log \lambda)$ under random noise of correlation ρ, and, for every alphabet, a constant fraction of the codeword if every error pattern up to a constant fraction of the length must be corrected, even when an unbounded adversary chooses it with the key. We call the last cost the robustness tax. As for non-committing encryption, adaptively secure PRCs thus need an a-priori bound on the number of openings, but each opening costs more, because of the tension between soundness and robustness: a sound key must reject almost all strings, yet accept a region around each uniformly random codeword. We show simple stateful codes meet these bounds and yield stateful PRCs with perfect bounded adaptive security.

Finally, unlike for non-committing encryption, a programmable random oracle cannot pay this cost under noise. Errors move the decoder's queries away from any programmed point, so for decoders with non-adaptive oracle queries, every opening must still be paid for in key bits or in programmed points that the decoder reads on a noticeable fraction of inputs.

Keywords: Pseudorandom Codes · Adaptive Corruption · Non-Committing Pseudorandom Codes · Lower Bounds · Isoperimetry

1 Introduction

Pseudorandom (error-correcting) codes (PRCs), introduced by Christ and Gunn [CG24], combine two properties in tension: codewords are computationally in-

distinguishable from uniform strings, yet decode correctly after a bounded number of errors. This makes PRCs a natural abstraction for robust and undetectable watermarking of generative models [CG24, GZS24, GM24], and they are now used as components of larger systems [LSS26]. Stronger security notions [AAC+25, DJK+26, CMY26] have been studied since, while black-box separations [GGW25, DMR25] and robustness limits [FGP+26] show that PRCs need coding structure.

In watermarking, the decoder key is held by a detector, and a detector may be compromised after watermarked content has already been published. Once the key is known, the codewords sent before the corruption can be decoded, so they no longer look random. Security under such *adaptive corruption* instead asks that the compromise reveal nothing beyond the messages. Following the simulation paradigm, the adversary's view, i.e., the earlier codewords together with the stolen key, should be indistinguishable from one in which the earlier codewords were truly uniform strings, the *dummies*, chosen without knowledge of the messages. A simulator that produces this view must, at corruption time, learn the messages and output a key that decodes each dummy to its message. We call this *opening* the dummies, and the key it outputs the *opened key*. Prior work [Cao26] formalizes this as a *decoder-non-committing* PRC (NC-PRC, Definition 5) with decoder-opening indistinguishability (IND-OPEN), and shows that adaptive UC security requires one, which LDPC-based PRCs [CG24] lack.

The same problem arises for non-committing encryption [CFGN96], where a simulated ciphertext must later be opened to any message. There, Nielsen [Nie02] showed that in the plain model the key must be at least as long as all messages it opens, so adaptively secure encryption needs an a-priori bound on the number of messages, and that a programmable random oracle removes this bound. For PRCs, two new constraints appear. First, a PRC must be *sound*: its key accepts only a small fraction of all strings, as otherwise random strings would be detected as watermarked. So even a key for zero-bit codes, which carry no message, must work to accept a uniform dummy. Second, a robust key must accept not only the dummy but also its noisy copies, or every word close to it. In this paper, we address the question:

How many key bits does it take to open a pseudorandom code,
and how does this cost depend on the errors the code must correct?

1.1 Our Contribution

Our main message is that adaptive security for PRCs is a storage problem: every opening costs as many key bits as the opened message is long, as for non-committing encryption [Nie02], plus a cost from the tension between soundness and robustness, which has no analogue for encryption: a sound key must reject almost all strings, yet accept a region around each uniformly random dummy. All our lower bounds are *information-theoretic*, so no computational assumption can lower this price. Table 1 summarizes our bounds, and Section 2 gives a technical overview.

Openings as Bounded-Use Codes. Every adaptively secure PRC yields a *q-use code* (Definition 9): its key is the opened key, which has the same length as the PRC's decoder key, and its codewords are the dummies, which are truly uniform. Whether a short key can decode them is thus a purely combinatorial question, so our bounds need no hardness assumption and hold *unconditionally*, hence also covering current PRC designs, which are computationally secure under hardness assumptions [CG24, GM24, AAC+25, DJK+26, CMY26] (Sections 2.1 and 4). Section 5.4 applies our bounds to the key lengths of these designs.

Lower Bounds on Key Length. Beyond the $\log|\mathcal{M}|$ bits of Nielsen's bound, each opening costs a second part from this tension between soundness and robustness, which grows with the set of words the opened key must decode around each codeword (Section 2.1, Section 5). The second part is (1) $\omega(\log\lambda)$ bits without errors (Theorem 1); (2) $\frac{1}{\rho^2}\cdot\omega(\log\lambda)$ bits under random noise of correlation ρ (Proposition 2); and (3) a constant fraction of the codeword under worst-case errors of weight below αn, for a constant $\alpha\in(0,\frac{1}{2})$ and codeword length n, specifically $\frac{2}{\ln 2}\alpha^2 n$ bits over every alphabet, and $(1-\mathsf{h}(\frac{1}{2}-\alpha))\,n$ over the binary one (Theorem 2), where h is the binary entropy function. Case (3) applies when these errors must be corrected even if an *unbounded* adversary chooses them with the key in hand after corruption. IND-OPEN cannot enforce this, so we require it explicitly.

Constructions with Matching Bounds. These bounds are met by simple *table-based codes*, whose key stores, for each codeword, some information about it (Sections 2.3 and 6). The *sketch table* (Construction 2) stores a short prefix of each codeword and accepts words whose prefix is close to it, which meets the bound without errors, and under random noise up to a constant factor (Lemma 5). The *full table* (Construction 1) stores each codeword exactly and accepts every word within distance αn of it, which corrects worst-case errors but costs the whole codeword (Lemma 5). The *covering code* (Construction 3) stores only the index of a nearby point of a fixed covering code, which meets the worst-case bound exactly for zero-bit codes ($\mathcal{M}=\{1\}$), and up to a factor of about 4 with messages (Lemma 6). Larger alphabets cost no more (Lemma 7). We turn these codes into stateful PRCs with perfect bounded adaptive security, which carry each message under a one-time pad (Construction 5 and Proposition 3). Under noise this costs about $\log|\mathcal{M}|/\rho^2$ bits per message, which is unavoidable for keys fixed before the messages (Proposition 4).

Random Oracles and Cost of Openings. Without noise, a programmable random oracle removes the cost, as for non-committing encryption [Nie02], where opening programs one fresh oracle point per codeword (Construction 6). Under noise it does not, since errors move the decoder's queries away from the programmed points. For decoders with non-adaptive queries, every opening must still be paid for in key bits or in *heavy* programmed points, which the decoder queries on a noticeable fraction of inputs, hence the oracle only relocates the cost (Theorem 3, and Sections 2.4 and 7).

2 Technical Overview

This section gives an overview of our work, and explains the idea behind each proof and construction, following the order of the paper: the reduction from openings to bounded-use codes (Section 2.1), the lower bounds on the key length (Section 2.2), the constructions that meet them (Section 2.3), and the random oracle model (Section 2.4). The formal results are in Section 4, Section 5, Section 6, and Section 7, respectively.

We here only fix the notation that the statements need (with the full list of notations given in Section 3). Let λ be the security parameter. The codeword space is Σ^n, where Σ is the alphabet, the message space is $\mathcal{M}$, and decoder keys are κ-bit strings. Throughout, errors are substitutions of symbols, measured by the Hamming distance $\mathsf{dist}(x, c)$, i.e., the number of positions on which a word x differs from a codeword c. An error of weight w turns c into a word x with $\mathsf{dist}(x, c) = w$. We say that a key is *sound* if it accepts at most a negligible fraction of all strings in Σ^n. The noise $\mathcal{N}_\rho$ replaces each bit of a codeword by a uniform bit with probability $1 - \rho$, independently, and $\mathsf{h}(x) = -x \log x - (1 - x) \log(1 - x)$ is the binary entropy function.

Table 1 lists, for each kind of error, our lower bound on the key length, and the construction that meets it.

2.1 Reduction from Openings to Bounded-Use Codes

A real PRC can use a short key for many codewords because its codewords look uniformly random only to someone without the key. Adaptive security breaks this. In the opened world of IND-OPEN (Definition 8, [Cao26]), i.e., the simulated view after corruption, the codewords are *dummies*, i.e., truly uniform strings that are sampled before their messages are known, and the opened key must still decode each dummy to its message. Since IND-OPEN only constrains efficient adversaries, the opened key inherits every property of the real key that an efficient adversary holding the key can test. This gives three properties (Lemma 1): the opened key decodes each dummy, decodes a noisy copy of each dummy, and rejects random strings.

These properties do not cover errors chosen *after* corruption, e.g., by someone who uses a stolen detector key to find the smallest change to a watermarked text that makes the detector no longer recognize the watermark. Such an adversary can aim its errors at the words near a dummy that the key decodes wrongly, so its errors are *worst-case*, and *every* word within some distance δ of each dummy must decode correctly. IND-OPEN cannot force this, since no efficient test can find a small set of wrongly decoded words inside a ball, so we require robustness against an *unbounded* adversary holding the key, which is the classical notion of worst-case robustness, and the one our lower bounds in Section 5 can price. Adaptive UC security [Cao26], which implies IND-OPEN, only guarantees that an *efficient* adversary holding the key cannot find such a word (but not that it does not exist). This computational requirement is weaker and more permissive,

Table 1: Lower bounds and matching constructions. For q-use codes, all entries are key bits per opened codeword *in addition to* the $\log|\mathcal{M}|$ bits for the message. For stateful PRCs, they are all key bits per codeword, including those for the message. Here $s = \omega(\log\lambda)$, the robustness radius is $\delta = \alpha n$, and lower-order terms are omitted ($O(\alpha\sqrt{n})$ in the lower bounds and $o(n)$ in the constructions for worst-case errors).

Errors	Lower Bound	Matching Construction	Cost of Construction
Plain model, q-use codes (Sections 5 and 6.1)			
None (exact decoding)	s [Theorem 1]	Sketch table [Construction 2]	$s + \log q$
Random noise $\mathcal{N}_\rho$	$(\frac{1}{\rho^2} - o(1))\, s$ [Proposition 2]	Sketch table [Construction 2]	$O(s/\rho^2)$
		Threshold code [Construction 4]	$(\frac{1}{\rho^2} + o(1))\, s$, inefficient
Worst-case, any Σ	$\max\{s,\ \frac{2}{\ln 2}\,\alpha^2 n\}$ [Theorem 2]	Full table [Construction 1]	$n\lceil\log\lvert\Sigma\rvert\rceil$
		Lifted covering code [Lemma 7]	$(1 - \mathsf{h}(\frac{1}{2} - \alpha))\, n$ for zero-bit codes and even $\lvert\Sigma\rvert$
Worst-case, $\Sigma = \{0,1\}$	$(1 - \mathsf{h}(\frac{1}{2} - \alpha))\, n$ [Theorem 2]	Covering code [Construction 3]	$(1 - \mathsf{h}(\frac{1}{2} - \alpha))\, n$ for zero-bit codes; $(1 - \mathsf{h}(\frac{1}{2} - 2\alpha))\, n$ with messages
Plain model, stateful PRCs (Section 6.2): all key bits per codeword			
None (exact decoding)	$\log\lvert\mathcal{M}\rvert + s$ [Corollary 1]	Table PRC, sketch test [Construction 5]	$\log\lvert\mathcal{M}\rvert + s$
Random noise $\mathcal{N}_\rho$	$\log\lvert\mathcal{M}\rvert + (\frac{1}{\rho^2} - o(1))\, s$ [Corollary 3]; and $(\frac{1}{\rho^2} - 1)\log\lvert\mathcal{M}\rvert$ for uniform codewords [Proposition 4]	Table PRC, sketch test [Construction 5]	$O(s/\rho^2) + \log\lvert\mathcal{M}\rvert/(1 - \mathsf{h}(\frac{1-\rho}{2}))$
Worst-case, any Σ	$\log\lvert\mathcal{M}\rvert + \max\{s,\ \frac{2}{\ln 2}\,\alpha^2 n\}$ [Corollary 4]	Table PRC, full test [Construction 5]	$n\lceil\log\lvert\Sigma\rvert\rceil$; no efficient PRC near the bound is known
Programmable random oracle model (Section 7): key bits in total			
None	no bound	Tagged oracle code [Construction 6]	λ, one oracle query
Random noise, non-adaptive decoder	$\Omega(\log\lvert\mathcal{M}\rvert + s)$ per opening, in key bits or heavy points [Theorem 3]	Oracle table [Construction 7]	λ, but the decoder reads n oracle bits per opening on every input
Random noise, adaptive decoder	open [Remark 4]	Oracle table [Construction 7]	as above

so a lower bound under it would apply to more PRCs. We leave it to future work whether our results extend to it (Remark 2).

In Section 4, we ignore how the opened key is found and keep only the dummies, the opened key, and the requirements on it. We abstract what remains as a *q-use code* $\mathsf{BUC}_q = (\mathtt{Gen}, \mathtt{Decode})$ (Definition 9):

- $(\mathsf{dk}, (c_1, \ldots, c_q)) \leftarrow \mathsf{BUC}_q.\mathtt{Gen}(1^\lambda, (m_1, \ldots, m_q))$ outputs a decoder key dk together with q codewords, one for each message;
- $\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, x)$ maps each codeword c_i to its message m_i.

Compared with a real PRC, the key dk may depend on the messages, and $\mathsf{BUC}_q.\mathtt{Gen}(\cdot)$ does not need to be efficient. This q-use code models the view after corruption, so each property of the opened key above becomes a property of the q-use code. In particular, the whole-ball requirement, which we call *statistical opening robustness* for the NC-PRC (Definition 15), becomes *statistical robustness* of the q-use code (Definition 14).

Proposition 1 (Informal). *Let* PRC *be correct and sound, with κ-bit decoder keys. If $\Sigma = \{0,1\}$, let it also be robust to $\mathcal{N}_\rho$. Let it admit a decoder-non-committing code* NC-PRC *with uniform dummies and Q-bounded* IND-OPEN*, i.e.,* IND-OPEN *when the decoder is corrupted after at most Q encodings. Then for every $q \le Q$, the dummies and the opened key form a q-use code BUC_q with key length κ that is perfectly pseudorandom, sound and correct, and, if $\Sigma = \{0,1\}$, also ρ-noise-robust. If* NC-PRC *is δ-statistically opening-robust, then BUC_q is δ-statistically robust.*

We give the formal statement in Proposition 1 of Section 4.3.

The codewords of this q-use code are the dummies, which are truly uniform rather than pseudorandom, even though PRC itself is only computationally pseudorandom. For the real PRC, hardness assumptions are what let a short key serve many codewords, but for truly uniform dummies they cannot help. Hence the only question is whether a short enough key exists among the 2^κ candidates, so every lower bound below is *unconditional*.

2.2 Lower Bounds on Key Length

In Section 5, we bound the key length of q-use codes. Each opening costs key bits in two parts. The *first part* is $\log|\mathcal{M}|$ bits for the opened message, as in Nielsen's bound [Nie02]. The *second part* comes from the tension between soundness and robustness (Section 1.1), and grows with the set of words around each codeword that the key must decode correctly: the codeword itself without errors, most noisy copies under random noise, and every nearby word under worst-case errors chosen by a key holder.

Exact Decoding and Random Noise. The first two cases follow from adaptive security alone: an *efficient* adversary holding the opened key can test both, and IND-OPEN forces the opened key to pass every such test that the real key passes.

Theorem 1 and Proposition 2 (Informal). *Let* BUC_q *be a perfectly pseudorandom, sound* q*-use code with key length* κ*, and let* $(\mathsf{dk}, (c_1, \ldots, c_q)) \leftarrow \mathsf{BUC}_q.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$*. Then:*

- *If* $\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, c_i) = m_i$ *for all* i*, except with negligible probability (i.e.,* BUC_q *is correct), then*

$$\kappa \geq q\,(\log|\mathcal{M}| + \omega(\log\lambda)).$$

- *If* $\Sigma = \{0,1\}$ *and* $\Pr[\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, c_i \oplus e) = m_i] \geq 1 - \mathsf{negl}(\lambda)$ *for each* i *and* $e \leftarrow\!\!\$\, \mathcal{N}_\rho(0^n)$ *(i.e.,* BUC_q *is* ρ*-noise-robust), then*

$$\kappa \geq q\left(\log|\mathcal{M}| + (1/\rho^2 - o(1)) \cdot \omega(\log\lambda)\right).$$

The same bounds hold with $q = Q$ *for every decoder-non-committing code* NC-PRC *with uniform dummies and* Q*-bounded* IND-OPEN*, the second if the PRC is robust to* $\mathcal{N}_\rho$ *(Corollaries 1 and 3).*

We give the formal statements in Theorem 1 and Proposition 2 of Section 5.2. The second part, $\omega(\log\lambda)$ bits per opening, has no analogue for encryption. It is the dominant cost for PRCs with short messages, and the only cost for zero-bit PRCs, whose message space is $\mathcal{M} = \{1\}$ and which are the main case in watermarking. In particular, no PRC with a fixed key length admits openings for an unbounded polynomial number of codewords.

Proof Idea: Counting Keys. The proof is a union bound over keys (Lemma 2). Fix a class $\mathcal{K}$ of keys, and suppose that every $K \in \mathcal{K}$ decodes a uniform dummy to a uniform message with probability at most β. The q dummies are independent, so some key in $\mathcal{K}$ decodes all of them correctly with probability at most $2^\kappa \beta^q$. If the opened key lies in $\mathcal{K}$ and decodes all dummies, this probability must be close to 1, so $\kappa \geq q\log(1/\beta)$.

The class $\mathcal{K}$ must be chosen with care. A key that accepts every string decodes every dummy, so counting all keys gives nothing. We therefore count only the keys that accept at most a $1/p$ fraction of all strings. The opened key is such a key for every $p = \mathrm{poly}(\lambda)$, since otherwise an efficient test would tell it apart from the real key, which is sound. The *cells* of such a key, i.e., the sets of strings that decode to each message, are disjoint and together cover at most a $1/p$ fraction of all strings. A uniform dummy therefore lands in the cell of a uniform message with probability $\beta \leq 1/(p\,|\mathcal{M}|)$, and each opening costs $\log|\mathcal{M}| + \log p$ bits. As this holds for every $p = \mathrm{poly}(\lambda)$, the second part is $\omega(\log\lambda)$.

Proof Idea: Noise and Hypercontractivity. Under noise, we count the dummies c whose noisy copies land in the cell of their message with probability at least $1/2$. Markov's inequality gives the same β up to a factor 2. The exact constant comes from hypercontractivity [Bon70, Bec75, O'D14]. The noise operator spreads every small set, so the noisy copies of c land in a set of measure $1/p$ with probability $1/2$ only if c itself lies in a set of measure about p^{-1/ρ^2}. Hence β drops to about $p^{-1/\rho^2}/|\mathcal{M}|$, and the second part grows by the factor $1/\rho^2$.

Worst-Case Errors and the Robustness Tax. Robust PRCs [CG24, AAC+25] already correct worst-case errors, but only errors chosen by an adversary *without* the key. Such an adversary cannot aim its errors, so a key may still decode a small, hidden set of words in each ball wrongly. After corruption, the adversary holds the key. An efficient adversary still cannot find these words, so IND-OPEN cannot rule them out, but an unbounded one can search the ball for them (Section 2.1). We therefore require that *every* word within distance δ of each codeword decodes correctly, so that even an unbounded adversary finds no wrongly decoded word, i.e., that the q-use code is *statistically robust* (Definition 14). For the q-use code obtained from an NC-PRC, this follows from statistical opening robustness of the NC-PRC (Proposition 1).

Theorem 2 (Informal). *Let* BUC_q *be a perfectly pseudorandom, sound,* δ*-statistically robust* q*-use code with key length* κ*, and let* $\delta = \alpha n$ *for a constant* α*. Then, for every alphabet* Σ*,*

$$\kappa \;\geq\; q\left(\log|\mathcal{M}| + \max\left\{\omega(\log\lambda),\; \frac{2}{\ln 2}\,\alpha^2 n - O(\alpha\sqrt{n})\right\}\right),$$

and, if $\Sigma = \{0,1\}$*,* $\kappa \geq q\left(\log|\mathcal{M}| + (1 - \mathsf{h}(\frac{1}{2} - \alpha))\, n - O(1)\right)$*.*

We give the formal statement in Theorem 2 of Section 5.3.

The second part is now a constant fraction of the codeword. We call it the *robustness tax*. This separates the two kinds of errors: random noise costs logarithmically many bits per opening, while worst-case errors cost linearly many. In Section 5.4, we apply both bounds to the key lengths of known PRCs.

Proof Idea: Interiors and Concentration of Measure. If every word within distance δ of a dummy must decode correctly, the dummy must lie in the δ-*interior* of its cell, i.e., the whole ball of radius δ around the dummy lies in the cell. In high dimension, almost every point of a set of measure at most $1/2$ is within distance about $\sqrt{n}$ of its complement. So the δ-interior of the accepting set of a sound key has measure at most $e^{-2(\delta-a)^2/n}$ for $a = \sqrt{(n/2)\ln 2}$, by McDiarmid's inequality [McD89] (Lemma 3). Over the binary alphabet, Harper's theorem [Har66] gives the optimal bound $2^{-n(1-\mathsf{h}(1/2-\alpha))}$ (Lemma 4). The interiors of different cells are disjoint, so the counting argument applies as before, with this measure in place of $1/p$.

2.3 Constructions with Matching Bounds

In Section 6, we show that these lower bounds are tight by giving constructions that match them. All our constructions are *table-based codes*, namely, the key stores, for each codeword, some information about it together with its message, and decoding looks for the entry that matches the received word. We first give q-use codes that meet the lower bounds for exact decoding and random noise, and for worst-case errors, and then turn them into stateful PRCs with perfect Q-bounded IND-OPEN (Section 6.2). In all q-use codes below, $\mathsf{BUC}_q.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$

first samples $c_1, \ldots, c_q \leftarrow\!\!\$\ \Sigma^n$ and outputs them as the codewords. The codes differ only in what the key stores and how $\mathsf{BUC}_q.\mathtt{Decode}(\cdot)$ uses it. Every decoder accepts a word x if it is close, in Hamming distance dist, to what the key stores about some codeword.

Lemmas 5 to 8 (Informal). *For every q, there is a perfectly pseudorandom, sound q-use code BUC_q with each of the following properties. Let $\delta = \alpha n$.*

- *If $\Sigma = \{0,1\}$, then for every $s = \omega(\log \lambda)$, BUC_q is correct and ρ-noise-robust with key length*
$$\kappa = q\left(\log|\mathcal{M}| + O(s/\rho^2)\right),$$
or $\kappa = q\left(\log|\mathcal{M}| + (1/\rho^2 + o(1))\, s\right)$ if $\mathsf{BUC}_q.\mathtt{Gen}(\cdot)$ need not be efficient.
- *If $|\Sigma|$ is even and $\mathcal{M} = \{1\}$, then BUC_q is δ-statistically robust with key length*
$$\kappa = q\left((1 - \mathsf{h}(\tfrac{1}{2} - \alpha))\, n + o(n)\right).$$
- *If $\Sigma = \{0,1\}$, then BUC_q is δ-statistically robust with key length*
$$\kappa = q\left(\log|\mathcal{M}| + (1 - \mathsf{h}(\tfrac{1}{2} - 2\alpha))\, n + o(n)\right).$$

We give the formal statements in Lemmas 5 to 8 of Section 6.1.

Sketch Table for Random Noise. Let $\Sigma = \{0,1\}$, and write $c|_{[\ell]}$ for the first ℓ bits of c. The *sketch table* (Construction 2) stores only these prefixes as the decoder key,
$$\mathsf{dk} = \left((c_1|_{[\ell]}, m_1), \ldots, (c_q|_{[\ell]}, m_q)\right),$$
and decodes by
$$\mathsf{STC}.\mathtt{Decode}(\mathsf{dk}, x) = m_i \quad \text{for the unique } i \text{ with } \mathsf{dist}(x|_{[\ell]}, c_i|_{[\ell]}) < \delta_\ell,$$
where $\delta_\ell = (2-\rho)\ell/4$, and $\mathsf{STC}.\mathtt{Decode}(\mathsf{dk}, x) = \perp$ if there is no such unique i.

The prefix of a noisy copy of c_i differs from $c_i|_{[\ell]}$ on about $(1-\rho)\ell/2$ positions, whereas the prefix of a random string differs on about $\ell/2$, and the radius δ_ℓ lies halfway. By Hoeffding's inequality, $\ell = O(\log p/\rho^2)$ bits tell the two apart except with probability $1/p$. This meets the second part of the lower bound for exact decoding, and for random noise up to a constant factor.

The constant factor can be removed, at the price of efficiency. The *threshold code* (Construction 4) stores, instead of the prefix itself, the index of a point that is slightly correlated with the prefix, taken from a fixed list of strings such that every prefix has such a point (a covering code, as below), and accepts the words that are still correlated with this point. Its key costs $(1/\rho^2 + o(1)) \log p$ bits per opening for soundness $1/p$, which matches Proposition 2, but generating the key is not known to be efficient (Lemma 8).

Full Table and Covering Code for Worst-Case Errors. The *full table* (Construction 1) stores each codeword exactly as the decoder key,

$$\mathsf{dk} = \big((c_1, m_1), \ldots, (c_q, m_q)\big),$$

and decodes by

$$\mathsf{FTC.Decode}(\mathsf{dk}, x) = m_i \quad \text{for the unique } i \text{ with } \mathsf{dist}(x, c_i) < \delta,$$

and $\mathsf{FTC.Decode}(\mathsf{dk}, x) = \perp$ if there is no such unique i. Since uniform codewords are far apart with high probability, the full table corrects every error of weight below δ.

The full table code, however, stores $n \log |\Sigma|$ bits per codeword, far above the robustness tax, although robustness only needs each codeword up to a small distance. The *covering code* (Construction 3) therefore stores each codeword only approximately. Let $\Sigma = \{0, 1\}$, and fix a list $\mathcal{C} = (w_1, \ldots, w_N)$ of strings such that every string in $\{0, 1\}^n$ is within distance r of some w_j. The covering code stores, for each codeword, the index of such a nearby string as the decoder key,

$$\mathsf{dk} = \big((j_1, m_1), \ldots, (j_q, m_q)\big) \quad \text{with } \mathsf{dist}(c_i, w_{j_i}) \le r,$$

and decodes by

$$\mathsf{CC.Decode}(\mathsf{dk}, x) = m_i \quad \text{for the smallest } i \text{ with } \mathsf{dist}(x, w_{j_i}) < \delta + r,$$

and $\mathsf{CC.Decode}(\mathsf{dk}, x) = \perp$ if there is no such i. By the triangle inequality, every word within distance δ of c_i is accepted. We choose r such that the decoding radius $\delta + r$ stays below $n/2$, so the code remains sound, and each index costs $\log N$ bits. If $\mathcal{M} = \{1\}$ and $\delta = \alpha n$, a covering code of the largest such radius has $\log N = (1 - \mathsf{h}(\frac{1}{2} - \alpha))\, n + o(n)$, which is exactly the robustness tax of Theorem 2. Seen this way, Theorem 2 is a converse for lossy compression of the dummies, and the covering code is the matching achievability.

For multi-bit codes, the key also stores each message, which costs exactly the first part. However, a worst-case error can move a word towards a codeword with a different message, so the neighborhoods of different codewords must be kept apart. This requires a finer covering code, whose index costs about four times the robustness tax for small α. Larger alphabets cost no more: mapping half of the symbols to 0 and the other half to 1 turns every even alphabet into the binary one without changing the key (Lemma 7).

Compiling Table-Based Codes to PRCs. A q-use code knows the messages when it generates its key, and can store them there. A real PRC cannot, since it fixes its key before any message is known. Our construction $\mathsf{PRC}_{\mathrm{tab}}$ (Construction 5) therefore turns a table-based q-use code into a PRC that carries each message inside the codeword, hidden under a one-time pad stored in the key. Each codeword $(u, w) \in \Sigma^{n_1} \times \Sigma^{n_2}$ consists of a *locator* u, which tells the decoder which table entry to use, and a *payload* w, which carries the message. The key

holds Q pairs (r_j, s_j) of uniform $r_j \leftarrow\!\$\ \Sigma^{n_1}$ and $s_j \leftarrow\!\$\ \Sigma^{n_2}$, which the encoder and the decoder share. Given an error-correcting code ECC, the t-th encoding and the decoding are

$$\mathsf{PRC}_{\mathrm{tab}}.\mathtt{Encode}(\mathsf{ek}, m) = \big(r_t,\ \mathsf{ECC}.\mathtt{Encode}(m) + s_t\big),$$
$$\mathsf{PRC}_{\mathrm{tab}}.\mathtt{Decode}(\mathsf{dk}, (u, w)) = \mathsf{ECC}.\mathtt{Decode}(w - s_j),$$

where j is the unique index whose locator r_j passes the full test $\mathsf{dist}(u, r_j) < \delta$ or the sketch test $\mathsf{dist}(u|_{[\ell]}, r_j|_{[\ell]}) < \delta_\ell$ of Constructions 1 and 2. With the sketch test, the key stores only the prefix $r_j|_{[\ell]}$. To open uniform dummies (u_i, w_i) to messages m_i, the simulator sets $r_i := u_i$ and $s_i := w_i - \mathsf{ECC}.\mathtt{Encode}(m_i)$. Since the dummies are uniform, these pairs are uniform and independent of the messages, exactly as in a real key. Hence $\mathsf{PRC}_{\mathrm{tab}}$ achieves perfect Q-bounded IND-OPEN (Proposition 3), i.e., it remains secure when its decoder key is corrupted after at most Q encodings, by making its first Q codewords perfectly uniform.

Under noise, this security has a price. The payload must also correct errors, so by the channel-coding theorem it needs about $\log|\mathcal{M}|/(1 - \mathsf{h}(\frac{1-\rho}{2})) \approx (2\ln 2/\rho^2)\log|\mathcal{M}|$ symbols, and so does the pad s_j, whereas a q-use code needs only $\log|\mathcal{M}|$ bits for the message. A cost of order $\log|\mathcal{M}|/\rho^2$ is unavoidable for every PRC whose key is fixed before the messages and whose first codewords are perfectly uniform:

Proposition 4 (Informal). *Let* PRC *be a secret-key stateful pseudorandom code over* $\Sigma = \{0,1\}$ *with* κ*-bit decoder keys, generated before the messages. Assume that, for all messages, the first* q *codewords of* PRC *are independent and uniform, and that each of them decodes to its message under* $\mathcal{N}_\rho$ *except with probability* ε*. Then*

$$\kappa \ \geq\ q\Big(\Big(\frac{1-\varepsilon}{\rho^2} - 1\Big)\log|\mathcal{M}| - \frac{1}{\rho^2}\Big).$$

We give the formal statement in Proposition 4 of Section 6.2. A uniform codeword alone carries no information about its message, so the message reaches the decoder only through the key and the noisy codeword together. By the strong data-processing inequality [AG76], the noise keeps only a ρ^2 fraction of this information, and the key must make up for the loss. With $q = Q$, the pads of Construction 5 are thus optimal up to the factor $2\ln 2$. An opened key, chosen after the messages, can store them directly, which is why Theorem 1 does not see this cost.

2.4 Random Oracles and Cost of Openings

In Section 7, we ask whether a *programmable* random oracle can pay the cost of openings instead of the key, as it does for non-committing encryption [Nie02]. Without noise, it can. In the tagged code (Construction 6), a codeword is $(u, H(s, u))$ for a random u and a λ-bit key s. To open dummies (u_i, w_i), the opener picks a fresh key $s^\star$ and programs $H(s^\star, u_i) := w_i$, one point per dummy, so the key stays λ bits however many codewords are opened.

Under Noise. An error in u moves the decoder's query away from the programmed point, so this no longer helps. Call an oracle point *heavy* if the decoder queries it on a noticeable fraction of all inputs, and *light* otherwise. The programmed points of the tagged code are light. We show that light points do not help under noise, at least for *non-adaptive* decoders, whose queries do not depend on earlier oracle answers.

Theorem 3 (Informal). *Let* PRC *be binary and robust to* $\mathcal{N}_\rho$ *for a constant* $\rho < 1$*, with a decoder that makes at most* T *non-adaptive queries with* ℓ_{out}*-bit answers, let the opener program at most* P *points, and let* $|\mathcal{M}| = \text{poly}(\lambda)$*. For every* $p = \text{poly}(\lambda)$ *and constant* $\theta \in (0,1)$ *there is* $\tau = 1/\text{poly}(\lambda)$ *such that, if at most* h *programmed points are queried by the opened decoder on at least a* τ *fraction of inputs, then*

$$\kappa + h\left(\ell_{\text{out}} + \log(T/\tau)\right) + \log(P+1) \;\geq\; (1-\theta)\, q\left(\log|\mathcal{M}| + \log p - O(1)\right).$$

We give the formal statement in Theorem 3 of Section 7.2. Heavy points act as key bits stored in the oracle, so the oracle only relocates the cost. Adaptive decoders are open (Remark 4).

3 Preliminaries

3.1 Notation

Throughout, Σ is a finite alphabet, n the codeword length, μ the uniform probability measure on Σ^n, and dist the Hamming distance, i.e., $\mathsf{dist}(x,c)$ is the number of coordinates on which $x, c \in \Sigma^n$ differ. It is our measure of errors throughout: a word x carries an error of weight $\mathsf{dist}(x,c)$ with respect to a codeword c. (It should not be confused with the statistical distance SD defined below.) We only consider substitution errors in the Hamming metric, and not insertions or deletions, as in edit-distance robust watermarks [GM24]. As in [Cao26], balls are open: $B_\delta(c) \coloneqq \{x : \mathsf{dist}(x,c) < \delta\}$. For a set $S \subseteq \Sigma^n$, its δ-*interior* is $\mathsf{Int}_\delta(S) \coloneqq \{c : B_\delta(c) \subseteq S\}$. Decoding is deterministic. For a key K of a code C and a message m, we write $A_K(m) \coloneqq \{x : \mathsf{C}.\mathtt{Decode}(K,x) = m\}$ and $A_K \coloneqq \bigcup_m A_K(m)$ for the accepting set, where C is clear from the context. Messages come from a finite message space $\mathcal{M}$ (where $\mathcal{M} = \{1\}$ is the zero-bit case) and decoder keys are κ-bit strings. For the binary alphabet, $\mathcal{N}_\rho$ denotes the binary symmetric channel that flips each bit independently with probability $(1-\rho)/2$, as in [GGW25, DMR25]. We write $f = \text{poly}(\lambda)$ if a function f of λ is bounded by a polynomial in λ, and $f = 1/\text{poly}(\lambda)$ if $1/f$ is. All parameters $n, \log|\mathcal{M}|, \kappa, q, Q$ are $\text{poly}(\lambda)$.

Statistical Distance. For a finite set Ω, we write $\mathcal{U}(\Omega)$ for the uniform distribution on Ω, and $x \leftarrow\!\!\$\ \Omega$ for sampling x uniformly from Ω. We also write $x \leftarrow\!\!\$\ D$ for sampling x from a distribution D, and $y \leftarrow A(x)$ for running a probabilistic

algorithm A on input x with fresh coins and output y. The *statistical distance* between random variables X and Y over Ω is

$$\mathsf{SD}(X,Y) \coloneqq \frac{1}{2}\sum_{\omega\in\Omega}\big|\Pr[X=\omega]-\Pr[Y=\omega]\big| = \max_{S\subseteq\Omega}\big|\Pr[X\in S]-\Pr[Y\in S]\big|.$$

We say that X is ε-*close* to Y if $\mathsf{SD}(X,Y)\le\varepsilon$. Statistical distance does not increase under any (possibly randomized) function applied to both arguments, so ε-closeness bounds the advantage of every distinguisher, efficient or not.

3.2 Pseudorandom Codes

Definition 1 (Pseudorandom Code [CG24]). *A* pseudorandom code *(PRC) is a triple* $\mathsf{PRC}=(\mathtt{KGen},\mathtt{Encode},\mathtt{Decode})$ *of PPT algorithms such that*

- $(\mathsf{ek},\mathsf{dk})\leftarrow\mathsf{PRC}.\mathtt{KGen}(1^\lambda)$*: on input the security parameter* λ*, outputs an encoding key* ek *and a decoder key* $\mathsf{dk}\in\{0,1\}^\kappa$*.*
- $c\leftarrow\mathsf{PRC}.\mathtt{Encode}(\mathsf{ek},m)$*: on input the encoding key* ek *and a message* $m\in\mathcal{M}$*, outputs a codeword* $c\in\Sigma^n$*. It may be randomized and stateful.*
- $m\coloneqq\mathsf{PRC}.\mathtt{Decode}(\mathsf{dk},x)$*: on input the decoder key* dk *and a word* $x\in\Sigma^n$*, deterministically outputs* $m\in\mathcal{M}\cup\{\perp\}$*.*

In a secret-key PRC both keys are secret, while in a public-key PRC ek is public. We recall the properties used in this paper. Below, $m_1,\dots,m_t$ is any sequence of messages of polynomial length, and $c_i\leftarrow\mathsf{PRC}.\mathtt{Encode}(\mathsf{ek},m_i)$ for $(\mathsf{ek},\mathsf{dk})\leftarrow\mathsf{PRC}.\mathtt{KGen}(1^\lambda)$.

Definition 2 (Correctness). PRC *is* correct *if*

$$\Pr[\forall i\in[t]:\mathsf{PRC}.\mathtt{Decode}(\mathsf{dk},c_i)=m_i]\ge 1-\mathsf{negl}(\lambda).$$

Definition 3 (Random-String Soundness). PRC *satisfies* random-string soundness *if*

$$\Pr_{\substack{(\mathsf{ek},\mathsf{dk})\leftarrow\mathsf{PRC}.\mathtt{KGen}(1^\lambda)\\ y\leftarrow\$\,\Sigma^n}}\big[\mathsf{PRC}.\mathtt{Decode}(\mathsf{dk},y)\neq\perp\big]\le\mathsf{negl}(\lambda).$$

Definition 4 (Robustness to $\mathcal{N}_\rho$). *Let* $\Sigma=\{0,1\}$ *and* $\rho\in(0,1]$. PRC *is* robust to $\mathcal{N}_\rho$ *if for every* $i\in[t]$*,*

$$\Pr[\mathsf{PRC}.\mathtt{Decode}(\mathsf{dk},c_i\oplus e)=m_i]\ge 1-\mathsf{negl}(\lambda),$$

where $e\leftarrow\!\!\$\,\mathcal{N}_\rho(0^n)$ *is independent of* $(\mathsf{ek},\mathsf{dk})$ *and the codewords.*

Robustness against adversarially chosen errors [CG24, AAC+25] is defined against adversaries that see codewords but not the decoder key. This distinction matters once the decoder key is exposed (Section 4.2).

3.3 Decoder-Non-Committing PRCs

We recall the decoder-non-committing PRCs of [Cao26], which capture security under adaptive corruption of the decoder. Before corruption, the adversary sees dummy codewords sampled without knowledge of the messages. After corruption, it receives a decoder key under which these dummies decode to the messages it asked for.

Definition 5 (Decoder-Non-Committing Code [Cao26]). *Let* $\mathsf{PRC} = (\mathtt{KGen}, \mathtt{Encode}, \mathtt{Decode})$ *be a pseudorandom code. We say that* PRC *admits a* decoder-non-committing pseudorandom code *if there exist PPT algorithms* $\mathsf{NC\text{-}PRC} = (\mathtt{SimKGen}, \mathtt{SimCode}, \mathtt{OpenDec})$ [1] *such that*

- $(\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{SimKGen}(1^\lambda)$*: on input the security parameter* λ*, outputs public information* pp *and an internal state* st*.*
- $c \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{SimCode}(\mathsf{pp})$*: on input the public parameter* pp*, outputs a dummy codeword* c*.*
- $\mathsf{dk}^\star \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{OpenDec}(\mathsf{st}, \mathcal{T})$*: on input the state* st *and any opening transcript* $\mathcal{T} = \{(m_i, c_i)\}_{i=1}^{q}$ *for some* $q \in \mathbb{N}$*, outputs a decoder key* $\mathsf{dk}^\star$*.*

In the secret-key case $\mathsf{pp} \coloneqq \bot$, while in the public-key case pp contains the public encoding key, i.e., $\mathsf{pp} \coloneqq \mathsf{ek}$.

Definition 6 (Simulated Nearby Code [Cao26]). *For an opening transcript* $\mathcal{T}$ *of message-codeword pairs* (m, c)*, we define the nearby-code set around* x *by*

$$\mathsf{SimNear}^{\delta}_{\mathcal{T}}(x) \coloneqq \{(m, c) \in \mathcal{T} : \mathsf{dist}(x, c) < \delta\}.$$

After corruption, the adversary can decode not only a dummy codeword c but also any perturbed word x with $\mathsf{dist}(x, c) < \delta$, and a consistent message must be decoded. This is captured by opening robustness.

Definition 7 (δ-Opening Robustness [Cao26]). *Let* $\mathsf{NC\text{-}PRC} = (\mathtt{SimKGen}, \mathtt{SimCode}, \mathtt{OpenDec})$ *be a decoder-non-committing pseudorandom code for* PRC*. We say that* $\mathsf{NC\text{-}PRC}$ *satisfies* δ-opening robustness *if, for every PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*,*

$$\Pr\left[\begin{array}{l|l} \exists!\,(m,\cdot) \in \mathsf{SimNear}^{\delta}_{\mathcal{T}}(x) & (\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{SimKGen}(1^\lambda), \\ \wedge\ \mathsf{PRC}.\mathtt{Decode}(\mathsf{dk}^\star, x) \neq m & \mathsf{st}_{\mathcal{A}} \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{Encode}}(\cdot)}(1^\lambda, \mathsf{pp}), \\ & \mathsf{dk}^\star \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{OpenDec}(\mathsf{st}, \mathcal{T}), \\ & x \leftarrow \mathcal{A}_2(\mathsf{st}_{\mathcal{A}}, \mathsf{dk}^\star) \end{array}\right] \leq \mathsf{negl}(\lambda),$$

where the oracle $\mathcal{O}_{\mathsf{Encode}}$ *invokes* $c \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{SimCode}(\mathsf{pp})$ *on each* m *adaptively queried by* $\mathcal{A}$*, responds* $\mathcal{A}$ *with* c*, and sets* $\mathcal{T} \coloneqq \mathcal{T} \cup \{(m, c)\}$ *with* $\mathcal{T}$ *initialized as* $\mathcal{T} \coloneqq \varnothing$*.*

[1] In [Cao26], $\mathsf{NC\text{-}PRC}.\mathtt{SimKGen}(\cdot)$ and $\mathsf{NC\text{-}PRC}.\mathtt{SimCode}(\cdot)$ also output trapdoors τ_k and τ_c that are passed to $\mathsf{NC\text{-}PRC}.\mathtt{OpenDec}(\cdot)$. They only identify the activations of the UC execution from which the NC-PRC is extracted and carry no cryptographic content. Since we work with games only, we omit them.

Opening robustness is only a consistency requirement: it does not guarantee that the opened key looks real. Decoder-opening indistinguishability requires that dummy codewords together with the opened key $\mathsf{dk}^\star$ be indistinguishable from honestly generated codewords together with the real decoder key (Fig. 1).

Game: $\mathbf{G}^{\text{IND-OPEN}}_{\mathsf{PRC},\mathsf{NC\text{-}PRC}}$

Procedure INIT

$b \leftarrow\!\$ \{0,1\}$
$\mathcal{T} := \varnothing$
$\mathsf{cor} := 0$
if $b = 0$ **then**
  $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{PRC.KGen}(1^\lambda)$
else
  $(\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{NC\text{-}PRC.SimKGen}(1^\lambda)$

Oracle $\mathcal{O}_{\mathsf{CorruptDec}}()$

if $\mathsf{cor} = 1$ **then**
  return $\perp$
$\mathsf{cor} := 1$
if $b = 0$ **then**
  return dk
else
  $\mathsf{dk}^\star \leftarrow \mathsf{NC\text{-}PRC.OpenDec}(\mathsf{st}, \mathcal{T})$
  return $\mathsf{dk}^\star$

Oracle $\mathcal{O}_{\mathsf{Encode}}(m)$

if $\mathsf{cor} = 1$ **then**
  return $\perp$
if $b = 0$ **then**
  $c \leftarrow \mathsf{PRC.Encode}(\mathsf{ek}, m)$
  $\mathcal{T} := \mathcal{T} \cup \{(m, c)\}$
else
  $c \leftarrow \mathsf{NC\text{-}PRC.SimCode}(\mathsf{pp})$
  $\mathcal{T} := \mathcal{T} \cup \{(m, c)\}$
return c

Procedure FIN

$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\{\mathsf{Encode},\mathsf{CorruptDec}\}}}(\mathsf{pp})$
return $b = b'$

Fig. 1: Decoder-opening indistinguishability (IND-OPEN) game for decoder-non-committing pseudorandom codes [Cao26]. The public value pp is given to the adversary. In public-key PRC, $\mathsf{pp} := \mathsf{ek}$, and in secret-key PRC, $\mathsf{pp} := \perp$.

Definition 8 (Decoder-Opening Indistinguishability). *Let* NC-PRC *be a decoder-non-committing code for a pseudorandom code* PRC*, as in [Cao26]. We say that* NC-PRC *satisfies* decoder-opening indistinguishability (IND-OPEN) *if, for every PPT adversary* $\mathcal{A}$*,*

$$\mathbf{Adv}^{\text{IND-OPEN}}_{\mathsf{PRC},\mathsf{NC\text{-}PRC}}(\mathcal{A}) := 2 \cdot \left| \Pr\left[\mathbf{G}^{\text{IND-OPEN}}_{\mathsf{PRC},\mathsf{NC\text{-}PRC}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda).$$

Throughout this work, we consider NC-PRCs with *uniform dummies*, that is, $\mathsf{NC\text{-}PRC.SimCode}(\mathsf{pp})$ outputs $c \leftarrow\!\$ \Sigma^n$, and IND-OPEN against adversaries that make at most Q encoding queries before corruption, which we call Q*-bounded* IND-OPEN.

4 Adaptive Openings as Bounded-Use Codes

An ordinary PRC can use a short decoder key for unboundedly many codewords, because its pseudorandomness is *computational* i.e., codewords only look random to someone without the key. Under *adaptive corruption*, the key may be exposed after some codewords have already been sent, and whoever holds it can decode

them. Adaptive security (IND-OPEN) hence requires that this view still be indistinguishable from one in which the earlier codewords were *truly* uniform strings i.e., the dummies, and the exposed key decodes each of them to its message. Since the dummies are sampled before the key and are truly random, no hardness assumption can help: the question is no longer whether an adversary can notice something, but whether a key that decodes all dummies to their messages exists at all. With κ-bit keys there are only 2^κ candidates, so this is a counting question.

In this section, we turn this question into one about a purely information-theoretic object, which we answer in Section 5.

4.1 Bounded-Use Codes

We first abstract adaptive openings as *bounded-use codes*. Ignoring how the opened key is found, what remains is a key together with q uniformly random strings that it decodes to prescribed messages. Definition 9 captures exactly this pair as a *q-use code*. Compared with a PRC, it has two relaxations. First, $\mathsf{BUC}_q.\mathtt{Gen}(\cdot)$ sees all messages before it outputs the key. Indeed, a real PRC fixes its key before any message is known, but an opened key is produced after the messages are fixed. Second, $\mathsf{BUC}_q.\mathtt{Gen}(\cdot)$ need not be efficient. We also note that neither relaxation weakens our lower bounds in Section 5, which count keys (Lemma 2) and never run $\mathsf{BUC}_q.\mathtt{Gen}(\cdot)$, so they hold even if the key is opened by an unbounded algorithm. Since a q-use code models the view after corruption, its key should be thought of as held by the adversary, which shapes how we define robustness below.

Definition 9 (q-Use Code). *Let Σ be a finite alphabet, n a length, $\mathcal{M}$ a finite message space, and κ a key length. A* q-use code *is a pair $\mathsf{BUC}_q = (\mathtt{Gen}, \mathtt{Decode})$ of algorithms:*

- $\mathsf{BUC}_q.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$*: on input messages $\boldsymbol{m} = (m_1, \ldots, m_q) \in \mathcal{M}^q$, outputs a decoder key $\mathsf{dk} \in \{0,1\}^\kappa$ and codewords $\boldsymbol{c} = (c_1, \ldots, c_q) \in (\Sigma^n)^q$;*
- $\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, x)$*: on input $x \in \Sigma^n$, deterministically outputs $m \in \mathcal{M} \cup \{\perp\}$.*

In Definitions 10 to 14, we let $\varepsilon = \varepsilon(\lambda) \in [0,1]$. When $\varepsilon = \mathsf{negl}(\lambda)$ we omit it. For example, we write *sound* for $\mathsf{negl}(\lambda)$-sound and *δ-statistically robust* for $(\delta, \mathsf{negl}(\lambda))$-statistically robust.

For a q-use code, pseudorandomness must be statistical, not computational. Since the key and the codewords are output together, there is no secret left for a computational notion to protect. For the codes obtained from openings, it is even perfect, since the opened strings are uniform (Proposition 1). A real PRC used for q encodings also gives a q-use code (run key generation and the first q encodings), but it satisfies Definition 10 only if its codewords are statistically close to uniform.

Definition 10 (Pseudorandomness). *A q-use code BUC_q is* ε-pseudorandom *if for every $\boldsymbol{m} \in \mathcal{M}^q$,*

$$\mathsf{SD}\big(\boldsymbol{c},\ \mathcal{U}((\Sigma^n)^q)\big) \le \varepsilon,$$

where $(\mathsf{dk}, \boldsymbol{c}) \leftarrow \mathsf{BUC}_q.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$. *It is* perfectly pseudorandom *if it is* 0*-pseudorandom, that is,* $c_1, \ldots, c_q$ *are independent and uniform on* Σ^n.

Soundness says that the key rejects random strings, as random-string soundness does for a PRC (Definition 3). This property is what makes openings expensive since a key that accepted most strings would decode every dummy at no cost.

Definition 11 (Soundness). *A* q*-use code* BUC_q *is* ε-sound *if for every* $\boldsymbol{m} \in \mathcal{M}^q$,

$$\Pr_{\substack{(\mathsf{dk},\boldsymbol{c})\leftarrow\mathsf{BUC}_q.\mathtt{Gen}(1^\lambda,\boldsymbol{m}) \\ y\leftarrow\$\Sigma^n}}\left[\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, y) \neq \perp\right] \leq \varepsilon,$$

where y *is uniform and independent of* dk. *Equivalently,* $\mathbb{E}[\mu(A_{\mathsf{dk}})] \leq \varepsilon$.

The remaining three properties ask the key to decode each codeword to its message, first without errors, then under random noise, and then under worst-case errors. Correctness is the requirement without errors.

Definition 12 (Correctness). *A* q*-use code* BUC_q *is* ε-correct *if for every* $\boldsymbol{m} \in \mathcal{M}^q$,

$$\Pr_{(\mathsf{dk},\boldsymbol{c})\leftarrow\mathsf{BUC}_q.\mathtt{Gen}(1^\lambda,\boldsymbol{m})}\left[\forall i \in [q] : \mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, c_i) = m_i\right] \geq 1 - \varepsilon.$$

Noise robustness asks the key to decode noisy copies of each codeword. The error is random noise, chosen independently of the key, so no adversary can aim it at the words that the key decodes wrongly. In Section 5, we show that this costs only $(1/\rho^2) \cdot \omega(\log \lambda)$ key bits per codeword beyond the message.

Definition 13 (Noise Robustness). *Let* $\Sigma = \{0, 1\}$ *and* $\rho \in (0, 1]$. *A* q*-use code* BUC_q *is* (ρ, ε)-noise-robust *if for every* $\boldsymbol{m} \in \mathcal{M}^q$ *and every* $i \in [q]$,

$$\Pr_{\substack{(\mathsf{dk},\boldsymbol{c})\leftarrow\mathsf{BUC}_q.\mathtt{Gen}(1^\lambda,\boldsymbol{m}) \\ e\leftarrow\$\mathcal{N}_\rho(0^n)}}\left[\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, c_i \oplus e) = m_i\right] \geq 1 - \varepsilon,$$

where e *is independent of* $(\mathsf{dk}, \boldsymbol{c})$.

The last property concerns worst-case errors. Since a q-use code models the view after corruption, such errors are chosen by an adversary that holds the key. Before corruption this would not matter: an adversary without the key sees only random-looking codewords and cannot aim its errors at the words that the key decodes wrongly, so standard robustness of PRCs [CG24, AAC+25], against adversaries without the key, suffices. After corruption, the adversary can search each ball for such words and choose its errors to land on them. For instance, someone who has stolen the detector's key may search for the smallest change to a watermarked text that the detector misses. To correct every error of weight below δ, every word within distance δ of each codeword must therefore decode correctly. We call this *statistical robustness*: no wrongly decoded word may exist,

so that even an *unbounded* adversary who knows the key cannot find one. We ask that no such word exist, rather than only that it be hard to find. The latter is a computational requirement: adaptive UC security, which implies IND-OPEN, guarantees that an efficient key holder cannot find such a word (the δ-opening robustness of Definition 7 [Cao26]). It is weaker than ours, since it allows wrongly decoded words that are merely hard to find, and whether our bounds extend to it is open (Remark 2). We use the statistical notion because it is the classical notion of worst-case robustness, which fits q-use codes as information-theoretic objects with no computational hardness assumption, and because it is what our counting argument can price (Theorem 2).

Definition 14 (Statistical Robustness). *Let $\delta > 0$. A q-use code BUC_q is* (δ, ε)-statistically robust *if for every $\boldsymbol{m} \in \mathcal{M}^q$,*

$$\Pr_{(\mathsf{dk},\boldsymbol{c}) \leftarrow \mathsf{BUC}_q.\mathsf{Gen}(1^\lambda, \boldsymbol{m})} \left[\forall i \in [q] : B_\delta(c_i) \subseteq A_{\mathsf{dk}}(m_i)\right] \geq 1 - \varepsilon,$$

that is, each c_i lies in $\mathsf{Int}_\delta(A_{\mathsf{dk}}(m_i))$.

Remark 1 (Unique-decoding regime). If the δ-balls of two codewords with different messages intersect, which happens exactly when the codewords are at distance at most $2\delta - 2$, no key is statistically robust for both. For uniform codewords over Σ this happens with negligible probability as long as $2\delta \leq (1 - 1/|\Sigma| - \epsilon)\, n$ for a constant $\epsilon > 0$ and $n = \omega(\log \lambda)$. We restrict statistical robustness to this regime. For zero-bit codes ($\mathcal{M} = \{1\}$) there is no such restriction.

4.2 Testing Opened Keys

We now determine what requirements that IND-OPEN forces on the opened key. The adversary may test the opened key in any way it could test a real key after corruption, so every test that a real key passes must also be passed by the opened key, on dummies that were sampled independently of their messages. Lemma 1 applies three such tests: decode each dummy, decode a noisy copy of each dummy, and reject random strings.

Lemma 1. *Let* PRC *be correct and satisfy random-string soundness, with decoder keys in $\{0,1\}^\kappa$. If $\Sigma = \{0,1\}$, let it also be robust to $\mathcal{N}_\rho$. Let* NC-PRC *be a decoder-non-committing pseudorandom code for* PRC *with uniform dummies and Q-bounded* IND-OPEN*. Fix $q \leq Q$ and messages $\boldsymbol{m}$. Let $c_1, \ldots, c_q$ be dummies and $\mathsf{dk}^\star$ the key opened on $\mathcal{T} = \{(m_i, c_i)\}_i$. Then:*

(1) $\Pr[\forall i : \mathsf{PRC.Decode}(\mathsf{dk}^\star, c_i) = m_i] \geq 1 - \mathsf{negl}(\lambda)$*;*

(2) if $\Sigma = \{0,1\}$, then $\Pr[\forall i : \mathsf{PRC.Decode}(\mathsf{dk}^\star, c_i \oplus e) = m_i] \geq 1 - \mathsf{negl}(\lambda)$*, where $e \leftarrow\!\!\$\, \mathcal{N}_\rho(0^n)$;*

(3) for every $p = \mathrm{poly}(\lambda)$, $\Pr[\mu(A_{\mathsf{dk}^\star}) > 1/p] \leq \mathsf{negl}(\lambda)$*, and hence* $\mathbb{E}[\mu(A_{\mathsf{dk}^\star})] \leq \mathsf{negl}(\lambda)$*.*

Proof. For (1), the adversary queries $m_1, \ldots, m_q$, corrupts the decoder, and outputs 1 iff every received codeword decodes to its message under the received key. In the real game it outputs 1 with probability at least $1 - q \cdot \mathsf{negl}(\lambda)$ by correctness, so by IND-OPEN it does so with probability $1 - \mathsf{negl}(\lambda)$ in the opened game. (2) is identical, with the adversary adding fresh noise $e \leftarrow\!\!\$\, \mathcal{N}_\rho(0^n)$ before decoding. In the real game this succeeds by robustness to $\mathcal{N}_\rho$.

For (3), fix p and let the adversary, after corruption, sample $t = 64\lambda p^2$ uniform strings, compute the fraction F accepted by the received key, and output 1 iff $F > 1/(2p)$. In the real game, random-string soundness and Markov's inequality give $\mu(A_{\mathsf{dk}}) \le 1/(4p)$ except with negligible probability, and then $F > 1/(2p)$ with probability at most $e^{-2t/(16p^2)} = e^{-8\lambda}$ by Hoeffding's inequality. In the opened game, whenever $\mu(A_{\mathsf{dk}^\star}) > 1/p$ we have $F \le 1/(2p)$ with probability at most $e^{-2t/(4p^2)}$. So the advantage is at least $\Pr[\mu(A_{\mathsf{dk}^\star}) > 1/p] - \mathsf{negl}(\lambda)$. Finally, $\mathbb{E}[\mu(A_{\mathsf{dk}^\star})] \le 1/p + \Pr[\mu(A_{\mathsf{dk}^\star}) > 1/p]$ for every $p = \mathrm{poly}(\lambda)$, so the expectation is negligible. □

By IND-OPEN, the opened key $\mathsf{dk}^\star$ inherits a property of the real key dk whenever (a) an efficient adversary holding the key can test the property, and (b) the real key dk has the property in the real game. Which robustness the opened key inherits therefore depends on who chooses the error.

- *(Random noise, independent of the key).* Both (a) and (b) hold, which gives Lemma 1(2). This is all that Corollary 3 needs.
- *(Errors chosen by an efficient key holder).* This is the δ-opening robustness of Definition 7. Condition (a) holds, but (b) requires the real PRC to be robust against adversaries that hold dk. Adversarial robustness of PRCs [CG24, AAC+25] does not provide this, since it is defined against adversaries without the key. Adaptive UC security does provide it: it yields δ-opening robustness [Cao26].
- *(Errors chosen by an unbounded key holder).* Here the whole δ-ball around each dummy must decode correctly. This never follows from IND-OPEN, because (a) fails: no efficient test can detect a small set of wrongly decoded words inside a ball.

For worst-case errors we require the last kind, for the reasons given before Definition 14: it is the classical notion of worst-case robustness, and it is what the robustness tax of Section 5.3 prices. Since IND-OPEN does not provide it, we state *statistical opening robustness* in Definition 15 as a separate assumption for NC-PRC, which Proposition 1 carries over to q-use codes as statistical robustness (Definition 14).

Definition 15 (Statistical Opening Robustness). *Let $\delta > 0$.* NC-PRC *is* (δ, ε)-statistically opening-robust *if for every $q \le Q$ and every $\boldsymbol{m} \in \mathcal{M}^q$, with dummies $c_1, \ldots, c_q$ and opened key $\mathsf{dk}^\star$ as above,*

$$\Pr\big[\forall i \in [q] : B_\delta(c_i) \subseteq A_{\mathsf{dk}^\star}(m_i)\big] \ge 1 - \varepsilon.$$

As for q-use codes, we write δ-statistically opening-robust *when $\varepsilon = \mathsf{negl}(\lambda)$.*

4.3 Reducing From Openings to Bounded-Use Codes

We now show the main step of our argument: the dummies together with the opened key form a *perfectly* pseudorandom q-use code with the same key length, whereas the PRC itself is only computationally pseudorandom. The real PRC's codewords look uniform only to an adversary without the key, which is what lets a short key serve many codewords. The extracted code's codewords are the dummies, which are *perfectly* uniform as random variables, and its key $\mathsf{dk}^\star$ must decode these truly random strings to the adversary's messages. Hardness assumptions cannot help such a key, because there is nothing left to hide. This is what makes the counting of Section 5 possible: every lower bound proved there for perfectly (or statistically) pseudorandom q-use codes applies to NC-PRCs.

Proposition 1. *Under the hypotheses of Lemma 1, for every $q \le Q$ the pair $\mathsf{BUC}_q = (\mathtt{Gen}, \mathtt{Decode})$ with*

$\mathsf{BUC}_q.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$

1 : $(\mathsf{pp}, \mathsf{st}) \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{SimKGen}(1^\lambda)$
2 : **for** $i \in [q]$ **do**
3 : $\quad c_i \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{SimCode}(\mathsf{pp})$
4 : $\mathsf{dk}^\star \leftarrow \mathsf{NC\text{-}PRC}.\mathtt{OpenDec}(\mathsf{st}, \{(m_i, c_i)\}_{i\in[q]})$
5 : **return** $(\mathsf{dk}^\star,\ (c_1, \ldots, c_q))$

$\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, x)$

1 : **return** $\mathsf{PRC}.\mathtt{Decode}(\mathsf{dk}, x)$

is a q-use code with key length κ that is perfectly *pseudorandom, even though* PRC *is only computationally pseudorandom, and that is sound and correct. If $\Sigma = \{0,1\}$, it is also ρ-noise-robust. If* NC-PRC *is δ-statistically opening-robust, then* BUC_q *is δ-statistically robust.*

Proof. The key of BUC_q is the opened key $\mathsf{dk}^\star$ and its codewords are the dummies, so each property of the opened world becomes a property of BUC_q.

- *(Pseudorandomness).* The dummies are independent and uniform, and independent of $\boldsymbol{m}$, so BUC_q is perfectly pseudorandom. The pseudorandomness of PRC plays no role here: BUC_q inherits the uniformity of the dummies, not that of the real codewords.
- *(Soundness).* $\mathsf{dk}^\star$ rejects random strings, by Lemma 1(3).
- *(Correctness).* $\mathsf{dk}^\star$ decodes each dummy to its message, by Lemma 1(1).
- *(Noise robustness).* If $\Sigma = \{0,1\}$, $\mathsf{dk}^\star$ decodes noisy copies of each dummy, by Lemma 1(2).
- *(Statistical robustness).* $\mathsf{dk}^\star$ decodes every word within distance δ of each dummy, which is exactly Definition 15.

Only statistical robustness is assumed. All other properties follow from IND-OPEN alone. □

5 Lower Bounds on Key Length

We now prove lower bounds on the key length of q-use codes. Since an adaptively secure PRC yields a q-use code with the same key length (Proposition 1), each bound is also a lower bound on the decoder key of the PRC. All lower bounds follow from one counting lemma (Section 5.1), applied to three increasingly demanding requirements on the opened key, one for each kind of error it must handle. Without errors, it must decode each dummy itself (exact decoding). Under random noise, chosen independently of the key, it must decode most noisy copies of each dummy (noisy decoding). Both cases are in Section 5.2. Under worst-case errors chosen by an adversary that holds the key, it must decode every word within distance δ of each dummy (whole-ball decoding, Section 5.3). The more words around a dummy the key must decode correctly, the rarer such dummies are for a fixed sound key, and the more key bits each opening costs. Exact and noisy decoding follow from adaptive security alone (Lemma 1), since an efficient adversary holding the opened key can test them. Whole-ball decoding cannot be tested efficiently, so we assume it: for NC-PRCs as statistical opening robustness (Definition 15), which makes the q-use code statistically robust (Definition 14).

5.1 A Counting Lemma

Fix a set $R \subseteq \{0,1\}^\kappa \times \Sigma^n \times \mathcal{M}$ and say that a key K *opens* a codeword c to a message m if $(K, c, m) \in R$. If every key in a class $\mathcal{K} \subseteq \{0,1\}^\kappa$ opens a uniformly random pair (c, m) with probability at most β, then, by a union bound over the at most 2^κ keys, some key in $\mathcal{K}$ opens q independent random pairs with probability at most $2^\kappa \beta^q$. Hence opening q pairs requires $\kappa \geq q \log(1/\beta)$, i.e., $\log(1/\beta)$ key bits per opened pair.

For non-committing encryption, the same counting with $\beta = 1/|\mathcal{M}|$ is Nielsen's bound $\kappa \geq q \log |\mathcal{M}|$ [Nie02]. For PRCs, however, soundness makes β smaller, because a sound key must also reject almost all random strings, and robustness makes it smaller still.

Lemma 2 (Counting). *Let $R \subseteq \{0,1\}^\kappa \times \Sigma^n \times \mathcal{M}$, let $\mathcal{K} \subseteq \{0,1\}^\kappa$, and let $\beta \geq \max_{K \in \mathcal{K}} \Pr_{c,m}[(K, c, m) \in R]$ for uniform $c \leftarrow\!\!\$\ \Sigma^n$ and $m \leftarrow\!\!\$\ \mathcal{M}$. Let BUC_q be an ε-pseudorandom q-use code with key length κ. Suppose that*

$$\Pr_{\substack{\boldsymbol{m} \leftarrow\!\!\$\, \mathcal{M}^q \\ (\mathsf{dk}, \boldsymbol{c}) \leftarrow \mathsf{BUC}_q.\mathtt{Gen}(1^\lambda, \boldsymbol{m})}} \left[\mathsf{dk} \in \mathcal{K} \ \wedge\ \forall i \in [q] : (\mathsf{dk}, c_i, m_i) \in R\right] \ \geq\ 1 - \gamma,$$

where $\boldsymbol{m} = (m_1, \ldots, m_q)$ and $\boldsymbol{c} = (c_1, \ldots, c_q)$. Then $2^\kappa \beta^q \geq 1 - \gamma - \varepsilon$, that is, $\kappa \geq q \log(1/\beta) - \log \frac{1}{1-\gamma-\varepsilon}$ whenever $\gamma + \varepsilon < 1$.

Proof. Let E be the set of pairs $(\boldsymbol{c}, \boldsymbol{m})$ for which some $K \in \mathcal{K}$ satisfies $(K, c_i, m_i) \in R$ for all i. The event in the hypothesis implies $(\boldsymbol{c}, \boldsymbol{m}) \in E$. For each fixed $\boldsymbol{m}$ the codewords are ε-close to uniform, so $\Pr[(\boldsymbol{c}, \boldsymbol{m}) \in E] \leq \Pr[(\boldsymbol{c}', \boldsymbol{m}) \in E] + \varepsilon$ for uniform $\boldsymbol{c}'$ independent of $\boldsymbol{m}$. By a union bound over $K \in \mathcal{K}$ and independence of the pairs (c'_i, m_i), $\Pr[(\boldsymbol{c}', \boldsymbol{m}) \in E] \leq |\mathcal{K}|\, \beta^q \leq 2^\kappa \beta^q$. □

The lemma never uses efficiency: the union bound ranges over all keys, so it holds even if $\mathsf{BUC}_q.\mathtt{Gen}(\cdot)$ (and hence $\mathsf{NC\text{-}PRC}.\mathtt{OpenDec}(\cdot)$) is unbounded.

5.2 Openings Cost Key Length

Our goal is a lower bound on κ. Lemma 2 lets us choose the class $\mathcal{K}$ freely and gives $\kappa \geq q \log(1/\beta)$, where β bounds how often a key in $\mathcal{K}$ opens a random pair: the smaller β is, the more key bits each opening costs. The choice of $\mathcal{K}$ matters, since a single key that opens random pairs often already makes β large. A large β means a weak bound: if some key opens most random pairs, then even a short key may open all q dummies, and the lemma cannot prove that the key must be long. Taking all keys is therefore too generous. A key that decodes every string to a fixed message m_0 opens a random pair whenever $m = m_0$, i.e., with probability $1/|\mathcal{M}|$, which gives only Nielsen's bound $\kappa \geq q \log |\mathcal{M}|$. For zero-bit codes, on the other hand, a key that accepts every string opens every pair, so $\beta = 1$ and the bound says nothing.

Both keys open pairs cheaply because they accept almost every string, and this is exactly what the key of a sound code cannot do. We therefore keep only the keys that accept at most a $1/p$ fraction of Σ^n:

$$\mathcal{K} = \mathcal{K}_p \coloneqq \{K \in \{0,1\}^\kappa : \mu(A_K) \leq 1/p\}.$$

Such a key accepts a random string with probability at most $1/p$, and must then also decode it to the right one of $|\mathcal{M}|$ messages. Under exact decoding it therefore opens a random pair with probability at most $1/(p\,|\mathcal{M}|)$ (see the proof of Theorem 1). Hence $\beta = 1/(p\,|\mathcal{M}|)$, and each opening costs $\log |\mathcal{M}| + \log p$ key bits: $\log |\mathcal{M}|$ bits to fix the message, as in Nielsen's bound, and $\log p$ bits to make a key that rejects almost everything accept the dummy.

It remains to show that the key output by a sound code lies in $\mathcal{K}_p$. Soundness (Definition 11) only says that keys accept few strings *on average*, $\mathbb{E}[\mu(A_{\mathsf{dk}})] \leq \varepsilon_{\mathrm{snd}}$, so a few keys could still accept many strings. By Markov's inequality, such keys are rare, that is, $\Pr[\mathsf{dk} \notin \mathcal{K}_p] \leq p\,\varepsilon_{\mathrm{snd}}$. If $\varepsilon_{\mathrm{snd}}$ is negligible, this is negligible for every $p = \mathrm{poly}(\lambda)$, so we may take $\log p = \omega(\log \lambda)$, which gives the corresponding term in Theorem 1.

Theorem 1 covers exact decoding and random noise $\mathcal{N}_\rho$, which is chosen independently of the key. Both are properties that an efficient adversary holding the opened key can test, and the real key passes these tests, so IND-OPEN forces them on the opened key (Lemma 1). Hence Corollary 1 below applies to every adaptively secure PRC, without any further assumption. Robustness of PRCs in the usual sense [CG24, AAC+25] also covers worst-case errors, but only errors chosen by an adversary that does not hold the key. After corruption, the adversary holds the key, which is the setting of Section 5.3.

Theorem 1 (Key Length). *Let BUC_q be an ε-pseudorandom, $\varepsilon_{\mathrm{snd}}$-sound q-use code with key length κ and message space $\mathcal{M}$. For every $p \geq 1$:*

1. *if* BUC_q *is* $\varepsilon_{\mathrm{cor}}$*-correct, then*

$$\kappa \geq q(\log|\mathcal{M}| + \log p) - \log \frac{1}{1 - p\varepsilon_{\mathrm{snd}} - \varepsilon_{\mathrm{cor}} - \varepsilon};$$

2. *if* BUC_q *is* $(\rho, \varepsilon_{\mathrm{rob}})$*-noise-robust, then*

$$\kappa \geq q(\log|\mathcal{M}| + \log p - 1) - \log \frac{1}{1 - p\varepsilon_{\mathrm{snd}} - 2q\varepsilon_{\mathrm{rob}} - \varepsilon}.$$

In particular, if $\varepsilon, \varepsilon_{\mathrm{snd}}, \varepsilon_{\mathrm{cor}}, \varepsilon_{\mathrm{rob}} \leq \mathsf{negl}(\lambda)$*, then* $\kappa \geq q\,(\log|\mathcal{M}| + \omega(\log\lambda))$.

Proof. Take $\mathcal{K} = \mathcal{K}_p$ in Lemma 2.

(Exact decoding). Let $R = \{(K, c, m) : \mathsf{BUC}_q.\mathtt{Decode}(K, c) = m\}$. For $K \in \mathcal{K}_p$ and independent uniform c, m,

$$\begin{aligned}\Pr[\mathsf{BUC}_q.\mathtt{Decode}(K, c) = m] &= \sum_{m' \in \mathcal{M}} \Pr_c[\mathsf{BUC}_q.\mathtt{Decode}(K, c) = m'] \cdot \frac{1}{|\mathcal{M}|} \\ &= \frac{\mu(A_K)}{|\mathcal{M}|} \leq \frac{1}{p\,|\mathcal{M}|},\end{aligned}$$

and the hypothesis of Lemma 2 holds with $\gamma = p\varepsilon_{\mathrm{snd}} + \varepsilon_{\mathrm{cor}}$.

(Noisy decoding). Let $R = \{(K, c, m) : \Pr_e[\mathsf{BUC}_q.\mathtt{Decode}(K, c \oplus e) = m] \geq 1/2\}$ with $e \leftarrow\!\!\$\ \mathcal{N}_\rho(0^n)$. By Markov's inequality applied to the conditional failure probability given (dk, c_i), each i violates R with probability at most $2\varepsilon_{\mathrm{rob}}$, so $\gamma = p\varepsilon_{\mathrm{snd}} + 2q\varepsilon_{\mathrm{rob}}$. For $K \in \mathcal{K}_p$ and each m, let f_m be the indicator of $A_K(m)$ and $T_\rho f_m(c) = \Pr_e[c \oplus e \in A_K(m)]$. Since $\mathbb{E}_c[T_\rho f_m(c)] = \mu(A_K(m))$, Markov's inequality gives $\Pr_c[T_\rho f_m(c) \geq 1/2] \leq 2\mu(A_K(m))$, so $\Pr_{c,m}[(K, c, m) \in R] \leq \frac{1}{|\mathcal{M}|}\sum_m 2\mu(A_K(m)) \leq \frac{2}{p\,|\mathcal{M}|}$.

(Asymptotics). If the errors are negligible, then for every $p = \mathrm{poly}(\lambda)$ the logarithmic term is at most 1 for large λ, so $\kappa/q - \log|\mathcal{M}| \geq \log p - 2$. As this holds for every $p = \mathrm{poly}(\lambda)$, $\kappa/q - \log|\mathcal{M}| = \omega(\log\lambda)$. □

Corollary 1. *Let* PRC *have* κ*-bit decoder keys and admit* NC-PRC *with uniform dummies and* Q*-bounded* IND-OPEN*, under the hypotheses of Lemma 1. Then* $\kappa \geq Q\,(\log|\mathcal{M}| + \omega(\log\lambda))$*. In particular, no PRC with a fixed key length admits openings for an unbounded polynomial number of codewords.*

Proof. Apply Theorem 1 to the Q-use code of Proposition 1. □

The bound has two parts. The term $Q\log|\mathcal{M}|$ is Nielsen's bound for non-committing encryption [Nie02], which counts message tuples. The term $Q\cdot\omega(\log\lambda)$ counts uniformly random codewords that a sound key must accept, and has *no analogue* for encryption. It is the only term for zero-bit PRCs, as used for watermarking, where Nielsen's bound is vacuous. As for non-committing encryption, adaptive security therefore needs an a-priori bound on the number of openings, and this holds without any computational assumption since the bound does not care how $\mathsf{NC\text{-}PRC}.\mathtt{OpenDec}(\cdot)$ finds the key.

Corollary 2. *For every decoding algorithm with κ-bit keys, every $p \geq 1$ with $p\,|\mathcal{M}| > 1$, and every $q \geq (\kappa+\lambda)/(\log|\mathcal{M}| + \log p)$,*

$$\Pr_{\boldsymbol{c},\boldsymbol{m}}\left[\exists K \in \mathcal{K}_p : \forall i \in [q] : \mathsf{BUC}_q.\mathtt{Decode}(K, c_i) = m_i\right] \leq 2^{-\lambda}$$

over uniform codewords $\boldsymbol{c}$ and uniform messages $\boldsymbol{m}$.

For LDPC-based PRCs, [Cao26, Lemma 2] shows the stronger statement quantified over *all* syntactically valid keys, for much smaller q. Corollary 2 shows that some such barrier holds for every PRC once q exceeds $\kappa/(\log|\mathcal{M}| + \log p)$.

Theorem 1 charges $\log p$ bits per noise-robust opening, as for exact decoding. Hypercontractivity [Bon70, Bec75, O'D14] sharpens this to the exact constant $1/\rho^2$.

Proposition 2. *Let $\Sigma = \{0,1\}$ and $\rho \in (0,1]$, and let BUC_q be a q-use code with key length κ and message space $\mathcal{M}$ that is ε-pseudorandom, $\varepsilon_{\mathrm{snd}}$-sound, and $(\rho, \varepsilon_{\mathrm{rob}})$-noise-robust. For every $p \geq 1$ and $r \geq 1$, with $s_r := r/(1+\rho^2(r-1))$,*

$$\kappa \geq q\big(\log|\mathcal{M}| + s_r \log p - r\big) - \log\frac{1}{1 - p\varepsilon_{\mathrm{snd}} - 2q\varepsilon_{\mathrm{rob}} - \varepsilon}.$$

In particular, for $p \geq 2$ and $r = \sqrt{\log p}$, each opening costs at least

$$\log|\mathcal{M}| + \frac{\log p}{\rho^2} - c_\rho\sqrt{\log p} \qquad \textit{bits, where } c_\rho := 1 + \frac{1-\rho^2}{\rho^4},$$

and if all errors are negligible, then $\kappa \geq q\left(\log|\mathcal{M}| + (1/\rho^2 - o(1))\,\omega(\log\lambda)\right)$.

Proof. We follow the noisy case of Theorem 1, with the same set $R = \{(K,c,m) : \Pr_e[\mathsf{BUC}_q.\mathtt{Decode}(K, c\oplus e) = m] \geq 1/2\}$, the same class $\mathcal{K} = \mathcal{K}_p$, and the same $\gamma = p\varepsilon_{\mathrm{snd}} + 2q\varepsilon_{\mathrm{rob}}$. Only the bound on β changes.

Fix $K \in \mathcal{K}_p$ and $m \in \mathcal{M}$, let f_m be the indicator of $A_K(m)$, and let $\mu_m := \mu(A_K(m))$. Then $T_\rho f_m(c) = \Pr_e[c \oplus e \in A_K(m)]$ is the noise operator applied to f_m. Let $t := 1 + \rho^2(r-1)$, so that $1 \leq t \leq r$ and, for $r > 1$, $\rho^2 = (t-1)/(r-1)$. The hypercontractive inequality [O'D14, Ch. 9] gives $\|T_\rho f_m\|_r \leq \|f_m\|_t = \mu_m^{1/t}$. For $r = 1$, this holds with equality, since T_ρ preserves the mean. By Markov's inequality applied to $(T_\rho f_m)^r$,

$$\Pr_c\left[T_\rho f_m(c) \geq 1/2\right] \leq 2^r\,\mathbb{E}_c\left[(T_\rho f_m)^r\right] = 2^r\|T_\rho f_m\|_r^r \leq 2^r\mu_m^{r/t} = 2^r\mu_m^{s_r}.$$

Since $(r-1)(1-\rho^2) \geq 0$, we have $s_r \geq 1$, and hence $\sum_m \mu_m^{s_r} \leq (\sum_m \mu_m)^{s_r} = \mu(A_K)^{s_r} \leq p^{-s_r}$. Averaging over a uniform m,

$$\Pr_{c,m}[(K,c,m) \in R] \leq \frac{1}{|\mathcal{M}|}\sum_m 2^r\mu_m^{s_r} \leq \frac{2^r}{p^{s_r}|\mathcal{M}|} =: \beta.$$

Lemma 2 gives the first bound, since $\log(1/\beta) = \log|\mathcal{M}| + s_r\log p - r$.

For the second bound, write $L = \log p$ and $r = \sqrt{L} \geq 1$. Since $s_r = \rho^{-2}\big(1 + \frac{1-\rho^2}{\rho^2 r}\big)^{-1} \geq \rho^{-2}\big(1 - \frac{1-\rho^2}{\rho^2 r}\big)$,

$$s_r L - r \;\geq\; \frac{L}{\rho^2} - \frac{(1-\rho^2)L}{\rho^4\sqrt{L}} - \sqrt{L} = \frac{L}{\rho^2} - c_\rho\sqrt{L}.$$

The asymptotic statement follows as in Theorem 1, since $c_\rho\sqrt{\log p} = o(\log p)$. □

For $r = 1$ we have $s_1 = 1$, and the proposition gives back the noisy case of Theorem 1. As for exact decoding, the bound carries over to adaptively secure PRCs.

Corollary 3. *Let $\Sigma = \{0,1\}$ and $\rho \in (0,1]$, and let* PRC *have κ-bit decoder keys and admit* NC-PRC *with uniform dummies and Q-bounded* IND-OPEN*, under the hypotheses of Lemma 1. Then, for every $p = \mathrm{poly}(\lambda)$ with $p \geq 2$ and all sufficiently large λ,*

$$\kappa \;\geq\; Q\left(\log|\mathcal{M}| + \frac{\log p}{\rho^2} - c_\rho\sqrt{\log p}\right) - 1,$$

with c_ρ as in Proposition 2. In particular, $\kappa \geq Q\left(\log|\mathcal{M}| + (1/\rho^2 - o(1))\,\omega(\log\lambda)\right)$.

Proof. Apply Proposition 2 with $r = \sqrt{\log p}$ to the Q-use code of Proposition 1, which is perfectly pseudorandom, sound and ρ-noise-robust with negligible errors. For every $p = \mathrm{poly}(\lambda)$, the logarithmic term of Proposition 2 is then at most 1 for large λ. □

Comparison with Prior Counting Arguments. The union bound over keys in Lemma 2 is not new. Nielsen uses it for non-committing encryption [Nie02], with $\beta = 1/|\mathcal{M}|$. Döttling, Müller and Rajasree [DMR25, Theorem 10] (reused in [DKRR26, Lemma 7.4]) and Garg, Gunn and Wang [GGW25, Lemma 1] use it, or its entropy version, to show that a PRC whose decoder needs no oracle cannot be pseudorandom against unbounded adversaries by giving a distinguisher that enumerates all keys to tell $O(\kappa + \lambda)$ codewords apart from random strings. Our use differs in three ways.

1. The bound applies to computationally secure PRCs, which include all current designs [CG24, GM24, AAC+25, DJK+26, CMY26]. Prior results run the counting on the real codewords of the PRC. Their distinguisher enumerates all 2^κ keys and checks whether one decodes all its samples. By Corollary 2, no sound key decodes more than $O(\kappa + \lambda)$ uniform strings, so this tells real codewords apart from random strings [DMR25, Theorem 10]. This distinguisher runs in exponential time, so it is not a valid attack on a PRC in the real world, where adversaries are efficient. It is valid relative to an oracle, where an adversary is bounded only in the number of queries it makes to the oracle, not in its running time. This is why those results rule out PRCs only relative to oracles. In contrast, we never need the real codewords to be uniform. Adaptive security already provides truly uniform strings, the

dummies of the opened world, and we run the counting on these dummies and the opened key. Since the dummies are uniform no matter how the PRC is built, the counting needs no assumption on the real codewords, which may be only computationally pseudorandom. Hence the counting is *unconditional*.

2. The bound measures the cost per codeword. Prior PRC results need only a constant β, which shows that $O(\kappa + \lambda)$ codewords suffice to distinguish. We take the smallest β each decoding requirement allows, which gives the number of bits per opening: $\log|\mathcal{M}| + \omega(\log\lambda)$ here, $(1/\rho^2)\log p$ under noise (Proposition 2), and $\Omega(\alpha^2 n)$ for whole balls (Theorem 2).
3. The bounds are tight (Section 6). Compared with Nielsen's bound, the new term comes from soundness, and from its tension with robustness once errors must be corrected. It is the only cost for zero-bit PRCs, where Nielsen's bound is vacuous, as discussed earlier.

An entropy version of the proof follows [GGW25, Lemma 1]. For a perfectly pseudorandom code and uniform messages, and ignoring the error events,

$$\begin{aligned} q\,(n\log|\Sigma| + \log|\mathcal{M}|) = H(\boldsymbol{c},\boldsymbol{m}) &\le \kappa + \sum_{i\in[q]} H(c_i, m_i \mid \mathsf{dk}) \\ &\le \kappa + q\,(n\log|\Sigma| - \log p). \end{aligned}$$

The last step holds because, given a key $\mathsf{dk} \in \mathcal{K}_p$, each pair (c_i, m_i) satisfies $\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, c_i) = m_i$, and there are only $|A_{\mathsf{dk}}| \le |\Sigma|^n/p$ such pairs. Note that the pairs must be counted jointly: given dk, the message m_i is no longer uniform, since the key is generated after the messages. We present the counting proof because it handles the error events directly, while the entropy version needs a Fano-type correction for them.

5.3 Robustness Tax

Theorem 1 charges each opening for placing a uniform dummy inside the accepting set of a sound key, which covers at most a $1/p$ fraction of Σ^n. Worst-case errors ask for more once the adversary holds the key. Robust PRCs [CG24, AAC+25] only correct errors chosen without the key, and may decode a small, hidden set of words in each ball wrongly. After corruption the adversary can search each ball for such words, so we ask that the whole δ-ball around each dummy decode correctly (Definition 14). Section 4.1 explains why we consider unbounded adversaries.

The dummy must then lie in the δ-interior of its cell, deep inside the accepting set rather than anywhere in it. In high dimension, almost every point of a set of measure at most $1/2$ is within distance about $\sqrt{n}$ of its complement (concentration of measure), so deep points are exponentially rare: the δ-interior has measure at most $e^{-\Theta(\delta^2/n)}$ (Lemma 3). A uniform dummy lands there only with this probability, so by Lemma 2 the key pays $\Theta(\delta^2/n)$ bits per opening. We call this extra cost, paid only because the openings must be robust, the *robustness tax*. For $\delta = \alpha n$ it is linear in n, a constant fraction of each codeword, whereas the cost in Theorem 1 is only logarithmic in λ.

Lemma 3 (Sparse Sets Have Small Interiors). *Let $S \subseteq \Sigma^n$ with $\mu(S) \le 1/2$, and let $a := \sqrt{(n/2)\ln 2}$. For every $\delta \ge a$,*

$$\mu(\mathsf{Int}_\delta(S)) \le e^{-2(\delta-a)^2/n}.$$

Proof. Let $Y := \Sigma^n \setminus S$ and $f(z) := \mathsf{dist}(z, Y)$, which changes by at most 1 when one coordinate of z changes. By McDiarmid's inequality [McD89], $\Pr[f \le \mathbb{E}f - t] \le e^{-2t^2/n}$ and $\Pr[f \ge \mathbb{E}f + t] \le e^{-2t^2/n}$ for $t \ge 0$. Since $f = 0$ exactly on Y and $\mu(Y) \ge 1/2$, the first inequality with $t = \mathbb{E}f$ gives $1/2 \le e^{-2(\mathbb{E}f)^2/n}$, i.e., $\mathbb{E}f \le a$. If $c \in \mathsf{Int}_\delta(S)$, then $B_\delta(c) \cap Y = \varnothing$, so $f(c) \ge \delta$. The second inequality with $t = \delta - \mathbb{E}f \ge \delta - a \ge 0$ gives the claim. □

With $\mu(Y) \ge 1 - 1/p$ instead of $1/2$, the same argument replaces a by $\sqrt{(n/2)\ln(p/(p-1))} \le \sqrt{n/p}$.

Over the binary alphabet, Harper's vertex-isoperimetric theorem [Har66] gives the optimal bound: Hamming balls minimize the measure of neighborhoods.

Lemma 4 (Binary Interiors). *Let $S \subseteq \{0,1\}^n$ with $\mu(S) \le 1/2$ and let $\delta \le n/2$ be a positive integer. Then*

$$\mu(\mathsf{Int}_\delta(S)) \le \Pr[\mathsf{Bin}(n, \tfrac{1}{2}) \ge \lceil n/2 \rceil + \delta - 1] \le 2^{-n\,(1 - \mathsf{h}(\frac{1}{2} - \frac{\delta-1}{n}))}.$$

Proof. Let $Y = \{0,1\}^n \setminus S$, so $|Y| \ge 2^{n-1} \ge |\bar{B}(\lceil n/2 \rceil - 1)|$, where $\bar{B}(r)$ is a closed Hamming ball of radius r. By Harper's theorem, the set Y_t of points within distance t of Y satisfies $|Y_t| \ge |\bar{B}(\lceil n/2 \rceil - 1 + t)|$. A point is outside $\mathsf{Int}_\delta(S)$ iff it is within distance $\delta - 1$ of Y, so $\mu(\mathsf{Int}_\delta(S)) \le 1 - |\bar{B}(\lceil n/2 \rceil + \delta - 2)|/2^n$, which is the stated binomial tail. The last step is the Chernoff bound $\Pr[\mathsf{Bin}(n, \frac{1}{2}) \ge an] \le 2^{-n(1-\mathsf{h}(a))}$ for $a \ge \frac{1}{2}$, together with $\mathsf{h}(\frac{1}{2} + x) = \mathsf{h}(\frac{1}{2} - x)$. □

Theorem 2 (Robustness Tax). *Let BUC_q be an ε-pseudorandom, $\varepsilon_{\mathrm{snd}}$-sound, $(\delta, \varepsilon_{\mathrm{rob}})$-statistically robust q-use code with key length κ and message space $\mathcal{M}$, and let $a = \sqrt{(n/2)\ln 2} \le \delta$. For every $p \ge 2$,*

$$\kappa \;\ge\; q\Big(\log|\mathcal{M}| + \max\Big\{\log p,\; \frac{2(\delta-a)^2}{n\ln 2}\Big\}\Big) - \log\frac{1}{1 - p\varepsilon_{\mathrm{snd}} - \varepsilon_{\mathrm{rob}} - \varepsilon}.$$

Over the binary alphabet, the term $\frac{2(\delta-a)^2}{n\ln 2}$ can be replaced by $n\,(1 - \mathsf{h}(\frac{1}{2} - \frac{\delta-1}{n}))$. In particular, if $\varepsilon, \varepsilon_{\mathrm{snd}}, \varepsilon_{\mathrm{rob}} \le \mathsf{negl}(\lambda)$ and $\delta = \alpha n$ for a constant α, each opening costs $\log|\mathcal{M}| + \frac{2}{\ln 2}\alpha^2 n - O(\alpha\sqrt{n})$ bits, and $\log|\mathcal{M}| + (1 - \mathsf{h}(\frac{1}{2} - \alpha))\,n - O(1)$ bits over the binary alphabet.

Proof. Take $\mathcal{K} = \mathcal{K}_p$ and $R = \{(K, c, m) : c \in \mathsf{Int}_\delta(A_K(m))\}$ in Lemma 2. The hypothesis holds with $\gamma = p\varepsilon_{\mathrm{snd}} + \varepsilon_{\mathrm{rob}}$. Fix $K \in \mathcal{K}_p$. The cells $A_K(m)$ are pairwise disjoint subsets of A_K, so their δ-interiors are pairwise disjoint subsets of $\mathsf{Int}_\delta(A_K)$. Hence

$$\Pr_{c,m}[(K, c, m) \in R] = \frac{1}{|\mathcal{M}|}\sum_{m \in \mathcal{M}} \mu(\mathsf{Int}_\delta(A_K(m))) \le \frac{\mu(\mathsf{Int}_\delta(A_K))}{|\mathcal{M}|}$$
$$\le \frac{\min\{1/p,\; e^{-2(\delta-a)^2/n}\}}{|\mathcal{M}|},$$

using $\mathsf{Int}_\delta(A_K) \subseteq A_K$ and Lemma 3 with $\mu(A_K) \le 1/p \le 1/2$. For the binary refinement use Lemma 4 instead. □

Corollary 4. *Under the hypotheses of Corollary 1, if the NC-PRC is also δ-statistically opening-robust and $\delta = \alpha n$, then*

$$\kappa \ge Q\Big(\log|\mathcal{M}| + \max\Big\{\omega(\log\lambda),\ \frac{2}{\ln 2}\alpha^2 n - O(\alpha\sqrt{n})\Big\}\Big),$$

with the binary refinement of Theorem 2.

For constant $\alpha = \delta/n$, an adaptively secure PRC with statistically robust openings must therefore store a constant fraction of every opened codeword. Section 6.1 shows that this is tight to leading order in α for zero-bit codes over every even alphabet, and, for small α, up to a factor of about 4 with messages: since $1 - \mathsf{h}(\frac{1}{2} - \alpha) = \frac{1}{\ln 2}(2\alpha^2 + \frac{4}{3}\alpha^4 + \cdots)$, the lower bound $\frac{2}{\ln 2}\alpha^2 n$ of Theorem 2 and the binary upper bound agree up to $O(\alpha^4 n)$.

Unlike Corollaries 1 and 3, Corollary 4 relies on statistical opening robustness (Definition 15), which IND-OPEN does not provide (Section 4.2).

Remark 2 (Robustness against efficient key holders). Adaptive UC security only yields the δ-opening robustness of Definition 7 [Cao26], which asks that no efficient key holder *find* a wrongly decoded word. This does not suffice for our argument: an opened key could decode a hidden set wrongly in every ball, so that no dummy lies in the δ-interior of its cell. Whether this weaker guarantee already forces a linear cost per opening, or whether such openings can cost $o(n)$ bits each, is open.

5.4 Consequences for Known PRCs

Every known PRC fixes the length κ of its decoder key at key generation, as a function of λ and n, before any (dummy) codeword is produced. The corollaries above therefore bound how many dummies such a PRC can open. We write $Q \lesssim X$ for $Q \le (1 + o(1))\,X$, i.e., we drop lower-order terms. A PRC with Q-bounded IND-OPEN, which constrains only efficient adversaries, satisfies

$$Q \ \lesssim\ \frac{\kappa}{\log|\mathcal{M}| + \omega(\log\lambda)}$$

by Corollary 1, and with $\omega(\log\lambda)$ replaced by $(1/\rho^2)\cdot\omega(\log\lambda)$ under noise by Corollary 3. If its openings must moreover be statistically robust with radius $\delta = \alpha n$ (Definition 15), i.e., withstand errors chosen by an unbounded key holder, which IND-OPEN cannot enforce, then Corollary 4 lowers this to

$$Q \ \lesssim\ \frac{\kappa}{\log|\mathcal{M}| + \frac{2}{\ln 2}\alpha^2 n},$$

for every alphabet.

In these bounds, κ is not necessarily the length of the key as written. For example, the LDPC decoder of [CG24] with key (P, z) computes only $P(x \oplus z) = Px \oplus Pz$. So the bound is determined by the parity-check matrix $P \in \{0,1\}^{r \times n}$, whose rows have $t = \Theta(\log n)$ ones each, and by the syndrome $Pz \in \{0,1\}^r$, rather than by the n-bit offset z. This gives $\kappa = O(r \log^2 n)$ instead of $n + O(r \log^2 n)$.

Table 2 instantiates these bounds for zero-bit constructions. They only say how many openings the key length leaves room for. A construction may open *far fewer* dummies because of its structure, for example if it has *targeted low opening capacity* (TLOC) [Cao26], as LDPC-based PRCs do. Without any structure, Corollary 2 shows that every PRC has a variant of TLOC once q exceeds $\kappa/(\log|\mathcal{M}| + \log p)$. The table gives the best limit we have in terms of key length.

Table 2: Upper bounds on the number Q of statistically robust openings of known zero-bit PRCs, from their (effective) key length κ, with n the codeword length and $\delta = \alpha n$. Without robustness, every row allows $Q \lesssim \kappa/\omega(\log \lambda)$. Keys: LDPC [CG24, Constr. 2], r checks of weight $\Theta(\log n)$ and the offset's syndrome, with $r = n^\varepsilon$ or $r = 0.99n$ [CG24, Thm. 1, 2]. Weak PRF [GM24, Alg. 1], a PRF key, an offset and a permutation of $[n]$. Permuted codes [CGG+25, Alg. 1, 3], a permutation of the positions of a code over $\mathbb{F}_q$, one permutation of $\mathbb{F}_q$ per position, and an offset. In the binary case the code has length λ, $q \approx \lambda$, and $n = \Theta(\lambda^{4/5} \log \lambda)$. Symbol-wise PRP [DMR25, Sec. 6], an encryption key and a PRP key.

Construction	Key length κ	Q, robust openings
LDPC [CG24]	$O(r \log^2 n)$	$O(r \log^2 n/(\alpha^2 n))$
Weak PRF [GM24]	$O(n \log n)$	$O(\log n/\alpha^2)$
Permuted, $\Sigma = \{0,1\}$ [CGG+25]	$\tilde{O}(n^{5/2})$	$\tilde{O}(n^{3/2}/\alpha^2)$
Permuted, $\Sigma = \mathbb{F}_q$ [CGG+25]	$O(n(\log n + q \log q))$	$O((\log n + q \log q)/\alpha^2)$
Symbol-wise PRP [DMR25]	$O(\lambda)$	0 if $n \gg \lambda/\alpha^2$

The table splits the constructions into two groups. In the first, the key leaves room for few robust openings or none. An LDPC-based PRC with $r = n^\varepsilon$ parity checks has $\kappa = O(n^\varepsilon \log^2 n)$, which is below the cost $\frac{2}{\ln 2}\alpha^2 n$ of a single robust opening once n is large. The same holds for an $O(\lambda)$-bit key once $n \gg \lambda/\alpha^2$. With $r = 0.99n$ parity checks, or with the $O(n \log n)$-bit keys of weak-PRF codes, only $O(\log^2 n/\alpha^2)$ or $O(\log n/\alpha^2)$ robust openings remain.

In the second group, the permuted codes of [CGG+25], binary or over a polynomially large field, have keys polynomially longer than a codeword, hence they allow polynomially many robust openings. Our bounds still apply, for instance $\tilde{O}(n^{3/2}/\alpha^2)$ in the binary case. They rule out only an unbounded number of openings. However, the structure is here again the tighter limit. Their per-symbol permutations can be programmed after the dummies are drawn, which [Cao26] sketches yields openings only while $Q^2 n \ll |\Sigma|$.

In every row, opening more dummies than the table allows requires a key that grows with Q, as in Construction 5.

Openings under a Public Encoding Key. In a public-key PRC [GG24, CG24, DJK+26, DFV26], the adversary receives the encoding key ek at the start, and in the opened world ek is output by $\mathsf{NC\text{-}PRC.SimKGen}(\cdot)$ before any dummy is drawn. After corruption, the adversary can encode fresh codewords under ek and check that $\mathsf{dk}^\star$ decodes them. It can also check that $\mathsf{dk}^\star$ contains ek whenever the real decoder key does. By IND-OPEN, the opened key must pass every such test that the real key passes. So it lies in a set $\mathcal{K}_{\mathsf{ek}}$ of keys determined by ek alone, and hence independent of the dummies. Running the counting of Lemma 2 conditioned on ek, with 2^κ replaced by $|\mathcal{K}_{\mathsf{ek}}|$, shows that the bounds above hold with κ replaced by $\log|\mathcal{K}_{\mathsf{ek}}|$. We only sketch this extension. The set $\mathcal{K}_{\mathsf{ek}}$ can be much smaller than $\{0,1\}^\kappa$. In the McEliece-based PRC of [DFV26], the decoder key (P, G', o) contains the encoding key (G', o), so only the permutation P of the n positions is free and $\log|\mathcal{K}_{\mathsf{ek}}| \le \log n! \approx n \log n$. Opening a public-key PRC is also harder for the simulator, since it must find a key that matches an encoding key fixed in advance. We leave a finer analysis of public-key constructions open.

6 Constructions with Matching Bounds

The bounds of Section 5 are lower bounds, i.e., every q-use code needs at least a certain key length. We now show that they are tight, by giving q-use codes whose keys are about that long (Section 6.1), and turn these codes into stateful PRCs with perfect bounded adaptive security (Section 6.2). All of them are based on tables: the key lists, for each codeword, some information about the codeword (the codeword itself, a prefix of it, or a nearby point of a fixed code) together with its message, and decoding looks for the entry that matches the received word. The codes correspond to two kinds of errors. *Worst-case errors* may be any pattern of weight below δ, chosen by an adversary who may hold the key. The full table code corrects these, and its refinement, the covering code of Construction 3, matches Theorem 2 for zero-bit codes. *Random noise* flips each bit independently, as in $\mathcal{N}_\rho$, and must only be corrected with high probability. The sketch table code corrects this and matches Theorem 1 and Proposition 2 up to a constant factor, which the threshold code of Construction 4 removes. Random noise is the weaker requirement: a code that corrects every error of a given weight also corrects random errors of that weight with high probability, but not conversely, since random errors are spread over all coordinates and cannot target the decoder's weak spots.

6.1 Bounded-Use Codes Meeting the Bounds

The full table code in Construction 1 stores every codeword in full, together with its message, and decodes a word to the message of the stored codeword within distance δ of it, if there is exactly one. Its codewords are sampled uniformly, so it is perfectly pseudorandom. Since the stored codewords are known exactly, it corrects worst-case errors: every error of weight below δ is corrected, and a random string is rejected unless it happens to fall within distance δ of one of the

q codewords. This is the robustness that Theorem 2 prices, but the full table pays $n \log |\Sigma|$ bits per opening, while Theorem 2 asks only for about $\frac{2}{\ln 2}\alpha^2 n$ when $\delta = \alpha n$. The covering code (Construction 3) closes this gap.

Construction 1 (Full Table Code)**.** *The* full table code $\mathsf{FTC} = (\mathtt{Gen}, \mathtt{Decode})$ *is defined as follows.*

- $\mathsf{FTC}.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$*: sample* $c_1, \ldots, c_q \leftarrow\!\!\$\ \Sigma^n$ *uniformly, and output the key* $\mathsf{dk} = ((c_1, m_1), \ldots, (c_q, m_q))$ *and the codewords* $(c_1, \ldots, c_q)$*.*
- $\mathsf{FTC}.\mathtt{Decode}(\mathsf{dk}, x)$*: if* i *is the unique index with* $\mathsf{dist}(x, c_i) < \delta$*, output* m_i*. Otherwise, output* $\perp$*.*

The full table code pays for a whole codeword per opening because it corrects worst-case errors. If the openings only have to survive random noise, Theorem 1 and Proposition 2 ask for far less, namely logarithmically many bits per opening, and the sketch table code in Construction 2 shows that this is achievable. A noisy copy of c differs from c on about a $(1-\rho)/2$ fraction of the coordinates, whereas an unrelated random string differs on about half of them, so already a short prefix of ℓ bits tells the two cases apart with high probability. The sketch table code therefore stores only the first ℓ bits of each codeword and, like the full table, accepts a word if it is close to a stored entry in Hamming distance: the distance on this prefix must be below the midpoint $\delta_\ell = (2-\rho)\ell/4$ of $(1-\rho)\ell/2$ and $\ell/2$. By a Chernoff bound, $\ell = O(s/\rho^2)$ bits suffice for error 2^{-s}. The price is that the code corrects only random noise, not worst-case errors: an adversary who knows the stored prefix can flip δ_ℓ of its bits and make the codeword undecodable.

Construction 2 (Sketch Table Code, Binary)**.** *Let* $\ell \le n$ *and* $\delta_\ell := (2-\rho)\ell/4$*, and write* $c|_{[\ell]}$ *for the first* ℓ *bits of* c*. The* sketch table code $\mathsf{STC} = (\mathtt{Gen}, \mathtt{Decode})$ *is defined as follows.*

- $\mathsf{STC}.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$*: sample* $c_1, \ldots, c_q \leftarrow\!\!\$\ \{0,1\}^n$ *uniformly, and output the key* $\mathsf{dk} = \big((c_1|_{[\ell]}, m_1), \ldots, (c_q|_{[\ell]}, m_q)\big)$ *and the codewords* $(c_1, \ldots, c_q)$*.*
- $\mathsf{STC}.\mathtt{Decode}(\mathsf{dk}, x)$*: if* i *is the unique index with* $\mathsf{dist}(x|_{[\ell]}, c_i|_{[\ell]}) < \delta_\ell$*, output* m_i*. Otherwise, output* $\perp$*.*

Lemma 5. *Write* $V_\delta := |B_\delta(0^n)|$*.*

1. *Construction 1 is perfectly pseudorandom,* ε_1*-sound, and* (δ, ε_2)*-statistically robust for* $\varepsilon_1 = qV_\delta/|\Sigma|^n$ *and* $\varepsilon_2 = q^2 V_{2\delta}/|\Sigma|^n$*, with key length* $q(n\lceil \log |\Sigma| \rceil + \lceil \log |\mathcal{M}| \rceil)$*.*
2. *Construction 2 is perfectly pseudorandom,* $(qe^{-\rho^2\ell/8})$*-sound,* $(q^2 e^{-\rho^2\ell/8})$*-correct, and* $(\rho, (q+1)e^{-\rho^2\ell/8})$*-noise-robust, with key length* $q(\ell + \lceil \log |\mathcal{M}| \rceil)$*. Choosing* $\ell = \lceil 8s/\rho^2 \rceil$ *for any* $s = \omega(\log \lambda)$ *makes all errors negligible, with key length* $q(\log |\mathcal{M}| + O(s/\rho^2))$*.*

Proof. Both codes output uniform, message-independent codewords. For the full table, a uniform y is accepted only if it lies in one of q balls of volume V_δ. If all pairwise distances are at least 2δ (failing with probability at most $q^2 V_{2\delta}/|\Sigma|^n$), every $x \in B_\delta(c_i)$ has c_i as its unique center within distance δ, so decodes to m_i.

For the sketch table, if x is uniform or $x = c_j$ for $j \neq i$, then $\mathsf{dist}(x|_{[\ell]}, c_i|_{[\ell]})$ is distributed as $\mathsf{Bin}(\ell, 1/2)$ and falls below $\delta_\ell = \ell/2 - \rho\ell/4$ with probability at most $e^{-\rho^2\ell/8}$ by Hoeffding's inequality. If $x = c_i \oplus e$ with $e \leftarrow\!\$\ \mathcal{N}_\rho(0^n)$, then $\mathsf{dist}(x|_{[\ell]}, c_i|_{[\ell]})\mathcal{S}_{\mathsf{Bin}}(\ell, (1-\rho)/2)$ reaches $\delta_\ell = (1-\rho)\ell/2 + \rho\ell/4$ with probability at most $e^{-\rho^2\ell/8}$, and for $j \neq i$ the distance to c_j is again $\mathsf{Bin}(\ell, 1/2)$. Union bounds give the claims: over the q entries for soundness, over the at most q^2 pairs of entries for correctness, and over the entry i and the $q-1$ other entries for noise robustness. □

For exact decoding, the radius $\delta_\ell = 1$ (an exact prefix match) gives soundness $q2^{-\ell}$ and correctness $q^2 2^{-\ell}$, so $\ell = \log p + \log q$ bits per opening suffice for soundness $1/p$, matching Theorem 1 up to the additive $\log q$.

The sketch table is not statistically robust: an adversary who knows the key flips $\delta_\ell \approx (2-\rho)\ell/4$ of the first ℓ coordinates, far fewer than δ for $\ell \ll \delta$, and the codeword is rejected. This gap is inherent: random noise costs $\log|\mathcal{M}| + (1/\rho^2) \cdot \omega(\log\lambda)$ bits per opening (Proposition 2), while worst-case errors cost a constant fraction of n (Theorem 2).

Exact Tightness for Worst-Case Errors. The full table in Construction 1 stores each dummy c_i exactly, which costs n bits per opening. Robustness does not need this much since the decoder only has to know c_i approximately. To make this precise, fix a *covering code* $\mathcal{C} \subseteq \{0,1\}^n$ of radius r, that is, a set of strings such that every $y \in \{0,1\}^n$ is within distance r of some $c' \in \mathcal{C}$, and list the elements of $\mathcal{C}$ in a fixed order. Instead of c_i, the key stores the position j_i in this list of a string $c'_i \in \mathcal{C}$ within distance r of c_i, hence taking $\lceil\log|\mathcal{C}|\rceil$ bits instead of n. The decoder now knows only c'_i, so it accepts every x within distance $\delta + r$ of c'_i. By the triangle inequality this includes the whole δ-ball around c_i. A larger r means a smaller $\mathcal{C}$ and hence a shorter key. Soundness limits how large r can be: it holds as long as $\delta + r$ stays below $n/2$, because a uniform string is then unlikely to be that close to any c'_i.

Construction 3 (Covering Code, Binary)**.** *Let $r \geq 0$ be an integer and let $\mathcal{C} = \{w_1, \ldots, w_{|\mathcal{C}|}\} \subseteq \{0,1\}^n$ be a code with covering radius r, i.e., every $y \in \{0,1\}^n$ has some w_j with $\mathsf{dist}(y, w_j) \leq r$. The* covering code $\mathsf{CC} = (\mathtt{Gen}, \mathtt{Decode})$ *is defined as follows.*

- $\mathsf{CC}.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$*: sample $c_1, \ldots, c_q \leftarrow\!\$\ \{0,1\}^n$ uniformly. For each i, let j_i be the smallest index with $\mathsf{dist}(c_i, w_{j_i}) \leq r$, so that j_i depends only on c_i, and let $c'_i := w_{j_i}$. Output the key $\mathsf{dk} = ((j_1, m_1), \ldots, (j_q, m_q))$ and the codewords $(c_1, \ldots, c_q)$.*
- $\mathsf{CC}.\mathtt{Decode}(\mathsf{dk}, x)$*: if i is the smallest index with $\mathsf{dist}(x, c'_i) < \delta + r$, output m_i. Otherwise, output $\perp$.*

Lemma 6. *Let $\delta = \alpha n$ and $\epsilon > 0$, and let $|\mathcal{C}| \leq n2^n/\bar{V}_r$, where $\bar{V}_r$ is the number of strings within distance r of 0^n. Such a code exists (see the proof and [CHLL97]). Construction 3 is perfectly pseudorandom, with key length $q(\lceil\log|\mathcal{C}|\rceil + \lceil\log|\mathcal{M}|\rceil)$, and $\log|\mathcal{C}| \leq n\big(1 - \mathsf{h}(r/n)\big) + 2\log(n+1)$ for $r \leq n/2$. Moreover:*

1. (Zero-bit). *If* $\mathcal{M} = \{1\}$ *and* $\delta + r \le (\frac{1}{2} - \epsilon)n$, *it is* $(qe^{-2\epsilon^2 n})$-*sound and* δ-*statistically robust with error* 0.
2. (Messages). *If* $2\delta + r \le (\frac{1}{2} - \epsilon)n$, *it is* $(qe^{-2\epsilon^2 n})$-*sound and* $(\delta, q^2 e^{-2\epsilon^2 n})$-*statistically robust.*

Proof. (Existence of $\mathcal{C}$*).* Let $M = \lceil n \ln 2 \cdot 2^n / \bar{V}_r \rceil$ and pick M uniform strings. A fixed y is not covered with probability $(1 - \bar{V}_r/2^n)^M < e^{-M\bar{V}_r/2^n} \le 2^{-n}$, so the expected number of uncovered strings is less than 1. Hence some choice covers all of $\{0,1\}^n$, and $M \le n2^n/\bar{V}_r$ for $n \ge 4$. Since $\bar{V}_r \ge \binom{n}{r} \ge 2^{n\mathsf{h}(r/n)}/(n+1)$ for $r \le n/2$, we get $\log |\mathcal{C}| \le \log n + n - n\mathsf{h}(r/n) + \log(n+1)$, which gives the stated bound.

(Pseudorandomness and soundness). The codewords are uniform, and the key only adds the indices j_i and the messages. Let y be uniform and independent of dk. For each i, $\mathsf{dist}(y, c_i')\mathcal{S}_{\mathsf{Bin}}(n, \frac{1}{2})$, so in both cases $\Pr[\mathsf{dist}(y, c_i') < \delta + r] \le \Pr[\mathsf{Bin}(n, \frac{1}{2}) < (\frac{1}{2} - \epsilon)n] \le e^{-2\epsilon^2 n}$ by Hoeffding's inequality, and a union bound over i gives soundness.

(Robustness). Let $x \in B_\delta(c_i)$. Then $\mathsf{dist}(x, c_i') \le \mathsf{dist}(x, c_i) + \mathsf{dist}(c_i, c_i') < \delta + r$, so x is accepted. In the zero-bit case every accepting index carries the same message, so x decodes correctly with probability 1. With messages, it remains to show that no $j \ne i$ has $\mathsf{dist}(x, c_j') < \delta + r$. The string c_j' depends only on c_j, so it is independent of c_i, and $\mathsf{dist}(c_i, c_j')\mathcal{S}_{\mathsf{Bin}}(n, \frac{1}{2})$. Except with probability $e^{-2\epsilon^2 n}$, $\mathsf{dist}(c_i, c_j') \ge (\frac{1}{2} - \epsilon)n \ge 2\delta + r$, and then $\mathsf{dist}(x, c_j') \ge \mathsf{dist}(c_i, c_j') - \mathsf{dist}(x, c_i) > \delta + r$ for every $x \in B_\delta(c_i)$. A union bound over the at most q^2 pairs (i, j) gives the claim. □

In the zero-bit case, take $r = \lfloor (\frac{1}{2} - \epsilon)n - \delta \rfloor$. Each opening then costs at most $n\big(1 - \mathsf{h}(\frac{1}{2} - \alpha - \epsilon)\big) + O(\log n)$ bits, while the binary bound of Theorem 2 asks for $n\big(1 - \mathsf{h}(\frac{1}{2} - \alpha)\big) - O(1)$. For every constant $\alpha < \frac{1}{2}$ the two differ by $O(\epsilon n + \log n)$. With $\epsilon = \sqrt{s/n}$ for $s = \omega(\log \lambda)$ and $s = o(n)$, soundness is negligible and the gap is $O(\sqrt{sn} + \log n) = o(n)$. So for zero-bit codes the covering code meets the binary bound of Theorem 2 up to $o(n)$ bits per opening, for every constant $\alpha < \frac{1}{2}$ and not only to leading order in α.

With messages, r must shrink to $(\frac{1}{2} - \epsilon)n - 2\delta$, because a worst-case error can move x by δ towards another center. Each opening then costs $\log |\mathcal{M}| + n\big(1 - \mathsf{h}(\frac{1}{2} - 2\alpha - \epsilon)\big) + O(\log n)$ bits, which is about $4 \cdot \frac{2}{\ln 2}\alpha^2 n$ for small α, a factor 4 above Theorem 2. The condition $2\delta < (\frac{1}{2} - \epsilon)n$ is the unique-decoding regime of Remark 1.

For Construction 3, the smaller radius is necessary. Two entries i and j have $\mathsf{dist}(c_i, c_j') \le (\frac{1}{2} + \epsilon)n$ except with probability $e^{-2\epsilon^2 n}$. Flipping $\delta - 1$ of the bits on which c_i and c_j' differ then gives a word $x \in B_\delta(c_i)$ with $\mathsf{dist}(x, c_j') \le (\frac{1}{2} + \epsilon)n - \delta + 1$, which is below $\delta + r$ as soon as $2\delta + r > (\frac{1}{2} + \epsilon)n + 1$. Such an x is accepted by both entries, and the decoder, which knows only c_i' and c_j', cannot tell which of them it belongs to. We do not know whether a different code avoids this loss. The lower bound cannot decide the question either, at least not through

Lemma 2, which only uses how often a *single* key opens a random pair. For $|\mathcal{M}| = 2$, the key whose two cells are the balls of radius $(\frac{1}{2} - \epsilon)n$ around 0^n and 1^n is sound and opens a random pair with probability $2^{-n(1-\mathsf{h}(\frac{1}{2}-\alpha-\epsilon))-O(\log n)}$, as much as Theorem 2 allows up to $2^{O(\epsilon n+\log n)}$. Such a key, however, cannot open q independent dummies at once, and a better lower bound would have to use this.

Remark 3 (Multi-bit codes under worst-case errors). It is open whether multi-bit q-use codes have to pay more than $\log|\mathcal{M}| + (1 - \mathsf{h}(\frac{1}{2} - \alpha))\, n + o(n)$ bits per opening for worst-case errors, or whether some code other than Construction 3 meets the bound of Theorem 2.

The covering code $\mathcal{C}$ can be a random linear code, which achieves the same size up to a poly(n) factor [CHLL97], and then $\mathsf{CC}.\mathtt{Decode}(\cdot)$ is efficient, since it only computes c'_i from j_i and q distances. $\mathsf{CC}.\mathtt{Gen}(\cdot)$, however, must round each dummy to $\mathcal{C}$, which is not known to be efficient. An efficient code meeting this bound is open.

Lifting to Other Alphabets. Binary constructions lift to every even alphabet without changing the key, so for these alphabets the robustness tax is the same as for binary.

Lemma 7. *Let $|\Sigma|$ be even and let $\pi : \Sigma \to \{0,1\}$ be balanced, i.e., each bit has exactly $|\Sigma|/2$ preimages, applied coordinatewise. Let $\mathsf{BUC}_q = (\mathtt{Gen}, \mathtt{Decode})$ be a binary q-use code. The* lifted code *$\mathsf{BUC}^{\pi}_q = (\mathtt{Gen}, \mathtt{Decode})$ over Σ is defined as follows.*

- $\mathsf{BUC}^{\pi}_q.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$*: run* $(\mathsf{dk}, (b_1, \ldots, b_q)) \leftarrow \mathsf{BUC}_q.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$*, sample each c_i uniformly from $\pi^{-1}(b_i)$, and output* dk *and* $(c_1, \ldots, c_q)$*.*
- $\mathsf{BUC}^{\pi}_q.\mathtt{Decode}(\mathsf{dk}, x)$*: output* $\mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, \pi(x))$*.*

If BUC_q is ε-pseudorandom, ε_1-sound, ε_2-correct, or (δ, ε_3)-statistically robust, then so is BUC^{π}_q, with the same key length.

Proof. (Pseudorandomness). The map $b \mapsto c$ applies the same randomized channel to each coordinate, and since π is balanced it sends a uniform bit to a uniform symbol. Hence it sends $\mathcal{U}((\{0,1\}^n)^q)$ to $\mathcal{U}((\Sigma^n)^q)$, and by the data-processing inequality $\mathsf{SD}(\boldsymbol{c}, \mathcal{U}) \le \mathsf{SD}(\boldsymbol{b}, \mathcal{U}) \le \varepsilon$.

(Soundness). A uniform $y \in \Sigma^n$, independent of dk, has uniform $\pi(y)$, still independent of dk, and $\mathsf{BUC}^{\pi}_q.\mathtt{Decode}(\mathsf{dk}, y) = \mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, \pi(y))$.

(Correctness). Follows from $\mathsf{BUC}^{\pi}_q.\mathtt{Decode}(\mathsf{dk}, c_i) = \mathsf{BUC}_q.\mathtt{Decode}(\mathsf{dk}, b_i)$.

(Robustness). Since π acts coordinatewise, $\mathsf{dist}(\pi(x), b_i) \le \mathsf{dist}(x, c_i)$, so π maps $B_\delta(c_i)$ into $B_\delta(b_i)$. Whenever every word of $B_\delta(b_i)$ decodes to m_i under BUC_q, every word of $B_\delta(c_i)$ decodes to m_i under BUC^{π}_q. □

With Lemma 6, every even alphabet therefore costs at most $(1-\mathsf{h}(\frac{1}{2}-\alpha))\,n+o(n)$ bits per zero-bit opening, which matches Theorem 2 to leading order in α. In particular, the full table, which stores $n\lceil\log|\Sigma|\rceil$ bits per opening, is far from optimal for large alphabets.

For odd $|\Sigma|$ no balanced π exists. A map with $(|\Sigma|+1)/2$ preimages of 0 biases $\pi(y)$ by $\beta = 1/(2|\Sigma|)$, and the covering code applied to $\pi(c_i)$ remains sound only if $\delta + r \le (\frac{1}{2}-\beta-\epsilon)n$. This gives $n(1-\mathsf{h}(\frac{1}{2}-\alpha-\beta-\epsilon))+O(\log n)$ bits per zero-bit opening, which matches Theorem 2 to leading order only when $|\Sigma|$ is large compared with $1/\alpha$. For small odd alphabets we do not know the exact constant.

Exact Constant for Random Noise. The sketch table stores raw bits and, even with the best radius, needs $\ell \approx (2\ln 2/\rho^2)\log p$ bits for soundness $1/p$ when ρ is small, a factor $2\ln 2$ above Proposition 2. Storing an index instead of raw bits closes this gap, as the covering code does for worst-case errors. The key stores, for each codeword, the index of a point of a covering code that is slightly correlated with the prefix of the codeword, at distance about $(\frac{1}{2}-b)\ell$ from it. A noisy copy of the codeword is then still correlated with this point, at distance about $(\frac{1}{2}-\rho b)\ell$, whereas a random string is at distance about $\ell/2$. The decoder therefore accepts the words within distance $(\frac{1}{2}-a)\ell$ of the point, for a threshold $a < \rho b$. Soundness $1/p$ needs $a \approx \sqrt{\ln p/(2\ell)}$, and with b slightly above a/ρ the index costs about $2b^2\ell/\ln 2 \approx (1/\rho^2)\log p$ bits.

Construction 4 (Threshold Code, Binary)**.** *Let $\ell \le n$ and $0 < a < b < \frac{1}{2}$, let $R := \lfloor(\frac{1}{2}-b)\ell\rfloor$, and let $\mathcal{C} = \{w_1,\dots,w_{|\mathcal{C}|}\} \subseteq \{0,1\}^\ell$ be a code with covering radius R. The* threshold code $\mathsf{TC} = (\mathtt{Gen},\mathtt{Decode})$ *is defined as follows.*

- $\mathsf{TC}.\mathtt{Gen}(1^\lambda, \boldsymbol{m})$*: sample $c_1,\dots,c_q \leftarrow\!\!\$\ \{0,1\}^n$ uniformly. For each i, let j_i be the smallest index with $\mathsf{dist}(c_i|_{[\ell]}, w_{j_i}) \le R$. Output the codewords $(c_1,\dots,c_q)$ and the key $\mathsf{dk} = ((j_1,m_1),\dots,(j_q,m_q))$.*
- $\mathsf{TC}.\mathtt{Decode}(\mathsf{dk}, x)$*: if i is the unique index with $\mathsf{dist}(x|_{[\ell]}, w_{j_i}) \le (\frac{1}{2}-a)\ell$, output m_i. Otherwise, output $\perp$.*

Lemma 8. *Let $\rho \in (0,1]$, $\ell \ge 4$, and $0 < a < \rho b$ with $b' := b + 1/\ell < \frac{1}{2}$, and let $|\mathcal{C}| \le \ell 2^\ell/\bar{V}_R$, where $\bar{V}_R$ is the number of strings in $\{0,1\}^\ell$ within distance R of 0^ℓ. Such a code exists by the proof of Lemma 6. Then Construction 4 is perfectly pseudorandom, $(qe^{-2a^2\ell})$-sound, $(q^2e^{-2a^2\ell})$-correct, and $(\rho,\ e^{-2(\rho b-a)^2\ell} + qe^{-2a^2\ell})$-noise-robust, with key length $q(\lceil\log|\mathcal{C}|\rceil + \lceil\log|\mathcal{M}|\rceil)$, where*

$$\log|\mathcal{C}| \;\le\; \frac{2b'^2\ell}{(1-4b'^2)\ln 2} + 2\log(\ell+1).$$

In particular, let ρ be constant, let p satisfy $\log p = \omega(\log\lambda)$, and let $n = \omega(\log p)$. Then, for suitable ℓ, a and b, the code is $(1/p)$-sound, correct and ρ-noise-robust, with key length

$$\kappa = q\left(\log|\mathcal{M}| + \left(\frac{1}{\rho^2} + o(1)\right)\log p\right).$$

Proof. The codewords are uniform and independent of the messages, so the code is perfectly pseudorandom. Write $D \coloneqq (\frac{1}{2} - a)\ell$ and $c'_i \coloneqq w_{j_i}$, and note that c'_i depends only on c_i.

(Soundness and correctness). If $y \in \{0,1\}^n$ is uniform and independent of c_i, then $\mathsf{dist}(y|_{[\ell]}, c'_i)\mathcal{S}_{\mathsf{Bin}}(\ell, \frac{1}{2})$, so $\Pr[\mathsf{dist}(y|_{[\ell]}, c'_i) \le D] \le e^{-2a^2\ell}$ by Hoeffding's inequality. A union bound over i gives soundness. For correctness, c_i passes the test for its own index, since $\mathsf{dist}(c_i|_{[\ell]}, c'_i) \le R \le D$, and it is uniform and independent of c_j for $j \ne i$, so it passes the test for j with probability at most $e^{-2a^2\ell}$. A union bound over the at most q^2 pairs gives correctness.

(Noise robustness). Let $x = c_i \oplus e$ with $e \leftarrow\!\$\, \mathcal{N}_\rho(0^n)$, and let $d \coloneqq \mathsf{dist}(c_i|_{[\ell]}, c'_i) \le R \le (\frac{1}{2} - b)\ell$. Given c_i, the distance $\mathsf{dist}(x|_{[\ell]}, c'_i)$ is a sum of ℓ independent bits: the $\ell - d$ positions on which c_i and c'_i agree each contribute 1 with probability $(1-\rho)/2$, and the d other positions with probability $(1+\rho)/2$. Its mean is

$$\frac{(1-\rho)\,\ell}{2} + \rho\, d \;\le\; \frac{(1-\rho)\,\ell}{2} + \rho\Big(\frac{1}{2} - b\Big)\ell = \Big(\frac{1}{2} - \rho b\Big)\ell,$$

so by Hoeffding's inequality it exceeds D with probability at most $e^{-2(\rho b - a)^2\ell}$. For $j \ne i$, the word x is uniform and independent of c_j, so it passes the test for j with probability at most $e^{-2a^2\ell}$. A union bound gives the claim.

(Key length). As in the proof of Lemma 6, $\log|\mathcal{C}| \le \ell\,(1 - \mathsf{h}(R/\ell)) + 2\log(\ell+1)$. Since $R/\ell \ge \frac{1}{2} - b'$ and

$$1 - \mathsf{h}\Big(\frac{1}{2} - b'\Big) = \frac{1}{\ln 2}\sum_{k\ge 1}\frac{(2b')^{2k}}{2k\,(2k-1)} \le \frac{1}{2\ln 2}\sum_{k\ge 1}(2b')^{2k} = \frac{2b'^2}{(1-4b'^2)\ln 2},$$

the stated bound on $\log|\mathcal{C}|$ follows.

(Asymptotics). Let $s \coloneqq \sqrt{\log\lambda \cdot \log p}$, so that $s = \omega(\log\lambda)$ and $s = o(\log p)$, and choose $\ell \le n$ with $\ell = \omega(\log p)$. Set

$$a \coloneqq \sqrt{\frac{\ln(pq)}{2\ell}} \qquad \text{and} \qquad b \coloneqq \frac{1}{\rho}\Big(a + \sqrt{\frac{s}{2\ell}}\Big).$$

Then $e^{-2a^2\ell} = 1/(pq)$ and $e^{-2(\rho b - a)^2\ell} = e^{-s}$, so the soundness error is $1/p$, the correctness error is q/p, and the noise-robustness error is at most $e^{-s} + 1/p$, all negligible. Since q is polynomial, $\ln q = o(\ln p)$, and hence

$$2b^2\ell = \frac{\big(\sqrt{\ln(pq)} + \sqrt{s}\big)^2}{\rho^2} = \frac{(1+o(1))\ln p}{\rho^2}.$$

Moreover $b = O(\sqrt{\log p/\ell}) = o(1)$, so $b' = o(1)$ and $2b'^2\ell = 2b^2\ell + 4b + 2/\ell = (1+o(1))\ln p/\rho^2$. Finally, $\log(\ell+1) = O(\log\lambda) = o(\log p)$, so $\log|\mathcal{C}| \le (1/\rho^2 + o(1))\log p$. □

Conversely, every $(1/p)$-sound, ρ-noise-robust q-use code pays $\log|\mathcal{M}|+(1/\rho^2-o(1))\log p$ bits per opening, by Proposition 2 applied with p/λ in place of p. So the threshold code is optimal up to a factor $1+o(1)$ in the second part. As for Construction 3, $\mathsf{TC.Decode}(\cdot)$ only computes distances, but $\mathsf{TC.Gen}(\cdot)$ must find a nearby point of $\mathcal{C}$, which is not known to be efficient.

6.2 From Table Codes to Adaptively Secure PRCs

A q-use code lets the key depend on the messages. A real stateful PRC cannot do this, because its decoder key is fixed before the messages are known. It, however, can carry the message in the codeword under a one-time pad stored in the key instead: each codeword consists of a uniform *locator* r_j, which the decoder finds with the full or sketch test, and a padded encoding of the message. Opening a uniform dummy then only requires writing the dummy's locator into the key and choosing the pad that turns the rest of the dummy into the encoding of the message.

Construction 5 (Table PRC)**.** *Identify Σ with $\mathbb{Z}_{|\Sigma|}$, let $n=n_1+n_2$, and parse words as $x=(u,w)\in\Sigma^{n_1}\times\Sigma^{n_2}$. Let $\mathsf{ECC}=(\mathtt{Encode},\mathtt{Decode})$ be an error-correcting code with $\mathsf{ECC.Encode}:\mathcal{M}\to\Sigma^{n_2}$ and $\mathsf{ECC.Decode}:\Sigma^{n_2}\to\mathcal{M}$ such that $\mathsf{ECC.Decode}(\mathsf{ECC.Encode}(m))=m$, and let $\mathsf{PRC}_0=(\mathtt{KGen},\mathtt{Encode},\mathtt{Decode})$ be any PRC over Σ^n with κ_0-bit decoder keys. The locator test $\mathsf{LocTest}(u,r)$ is either the* full test $\mathsf{dist}(u,r)<\delta$*, or, for $\Sigma=\{0,1\}$, the* sketch test $\mathsf{dist}(u|_{[\ell]},r|_{[\ell]})<\delta_\ell$ *of Construction 2. With the sketch test, dk and $\mathsf{dk}^\star$ store only $r_j|_{[\ell]}$ instead of r_j. The secret-key stateful pseudorandom code $\mathsf{PRC}_{\mathrm{tab}}=(\mathtt{KGen},\mathtt{Encode},\mathtt{Decode})$ is defined as follows.*

- $\mathsf{PRC}_{\mathrm{tab}}.\mathtt{KGen}(1^\lambda)$*: sample $r_j\leftarrow\!\!\$\ \Sigma^{n_1}$ and $s_j\leftarrow\!\!\$\ \Sigma^{n_2}$ for $j\in[Q]$, and $(\mathsf{ek}_0,\mathsf{dk}_0)\leftarrow\mathsf{PRC}_0.\mathtt{KGen}(1^\lambda)$. Output $\mathsf{ek}=((r_j,s_j)_{j\in[Q]},\mathsf{ek}_0)$ with a counter $t:=0$, and $\mathsf{dk}=((r_j,s_j)_{j\in[Q]},\mathsf{dk}_0)$.*
- $\mathsf{PRC}_{\mathrm{tab}}.\mathtt{Encode}(\mathsf{ek},m)$*: set $t:=t+1$. If $t\le Q$, output $(r_t,\ \mathsf{ECC.Encode}(m)+s_t)$. Otherwise, output $\mathsf{PRC}_0.\mathtt{Encode}(\mathsf{ek}_0,m)$.*
- $\mathsf{PRC}_{\mathrm{tab}}.\mathtt{Decode}(\mathsf{dk},(u,w))$*: if exactly one $j\in[Q]$ passes $\mathsf{LocTest}(u,r_j)$, output $\mathsf{ECC.Decode}(w-s_j)$. If none does, output $\mathsf{PRC}_0.\mathtt{Decode}(\mathsf{dk}_0,(u,w))$. Otherwise, output $\perp$.*

The decoder-non-committing code $\mathsf{NC\text{-}PRC}_{\mathrm{tab}}=(\mathtt{SimKGen},\mathtt{SimCode},\mathtt{OpenDec})$ is defined as follows.

- $\mathsf{NC\text{-}PRC}_{\mathrm{tab}}.\mathtt{SimKGen}(1^\lambda)$*: sample $(\mathsf{ek}_0,\mathsf{dk}_0)\leftarrow\mathsf{PRC}_0.\mathtt{KGen}(1^\lambda)$ and output $\mathsf{pp}:=\perp$ and $\mathsf{st}:=(\mathsf{ek}_0,\mathsf{dk}_0)$.*
- $\mathsf{NC\text{-}PRC}_{\mathrm{tab}}.\mathtt{SimCode}(\mathsf{pp})$*: output a uniform $c\leftarrow\!\!\$\ \Sigma^n$.*
- $\mathsf{NC\text{-}PRC}_{\mathrm{tab}}.\mathtt{OpenDec}(\mathsf{st},\mathcal{T})$*: for the transcript $\mathcal{T}=((m_i,(u_i,w_i)))_{i\in[q]}$ in query order, set $r_i:=u_i$ and $s_i:=w_i-\mathsf{ECC.Encode}(m_i)$ for $i\le q$, sample fresh r_j,s_j for $q<j\le Q$, and output $\mathsf{dk}^\star=((r_j,s_j)_{j\in[Q]},\mathsf{dk}_0)$.*

Proposition 3. *The code $\mathsf{NC\text{-}PRC}_{\mathrm{tab}}$ has uniform dummies. Its Q-bounded IND-OPEN is perfect, i.e., every adversary, even an unbounded one, has advantage 0. If PRC_0 is pseudorandom, so is $\mathsf{PRC}_{\mathrm{tab}}$, and its first Q codewords are perfectly uniform. Let t be the number of encodings and $V_d^{(n_1)}:=|B_d(0^{n_1})|$.*

1. (Full test). *Suppose* $\mathsf{ECC.Decode}(\mathsf{ECC.Encode}(m)+e)=m$ *whenever* e *has weight below* δ. *Then, up to an additive error* $\varepsilon_{\text{full}} \coloneqq (Q+t+1)^2 V_{2\delta}^{(n_1)}/|\Sigma|^{n_1}$ *beyond the errors of* PRC_0, $\mathsf{PRC}_{\text{tab}}$ *is correct and sound, and every word within distance* δ *of one of its first* Q *codewords decodes to that codeword's message.* $\mathsf{NC\text{-}PRC}_{\text{tab}}$ *is* $(\delta, \varepsilon_{\text{full}})$*-statistically opening-robust. The decoder key has* $Q\, n\lceil \log|\Sigma|\rceil + \kappa_0$ *bits.*
2. (Sketch test, $\Sigma=\{0,1\}$). *Suppose* $\mathsf{ECC.Decode}(\mathsf{ECC.Encode}(m)\oplus e)=m$ *except with probability* $\varepsilon_{\mathsf{ECC}}$ *for* $e \leftarrow\!\$\, \mathcal{N}_\rho(0^{n_2})$. *Then, up to an additive error* $\varepsilon_{\text{sk}} \coloneqq (Q+t+1)^2 e^{-\rho^2\ell/8} + \varepsilon_{\mathsf{ECC}}$ *beyond the errors of* PRC_0, $\mathsf{PRC}_{\text{tab}}$ *is correct, sound and robust to* $\mathcal{N}_\rho$, *and so is the opened key on the dummies. The decoder key has* $Q(\ell+n_2)+\kappa_0$ *bits.*

If the encoder stops after Q *encodings,* PRC_0 *can be omitted, and all statements hold unconditionally with* $t \le Q$ *and* $\kappa_0 = 0$.

Proof. (IND-OPEN). Fix messages $m_1, \dots, m_q$ for $q \le Q$, possibly chosen adaptively. In the real game, $c_i = (r_i, \mathsf{ECC.Encode}(m_i) + s_i)$ for fresh uniform (r_i, s_i). For fixed m_i the map $(r_i, s_i) \mapsto c_i$ is a bijection of $\Sigma^{n_1} \times \Sigma^{n_2}$, so c_i is uniform and independent of the adversary's view so far, and $(r_i, s_i) = (u_i, w_i - \mathsf{ECC.Encode}(m_i))$ is determined by c_i and m_i. The entries for $j > q$ and dk_0 are independent of the codewords. In the ideal game, the dummies are uniform and independent, and $\mathsf{dk}^\star$ is formed by the same formula, with the same distribution for the remaining entries. Hence the adversary's views in the two games are identically distributed. This also gives pseudorandomness: the first Q codewords are perfectly uniform, and later codewords come from PRC_0, whose keys are independent of the table.

(Full test). Let G be the event that the r_j are pairwise at distance at least 2δ, and that no base codeword and no test string y lies within distance δ of any r_j. The base codewords and y are independent of the r_j, so a union bound gives $\Pr[\neg G] \le (Q^2 + (t+1)Q) V_{2\delta}^{(n_1)}/|\Sigma|^{n_1} \le \varepsilon_{\text{full}}$. Assume G. If $\mathsf{dist}(x, c_j) < \delta$ for a table codeword c_j, then $\mathsf{dist}(u, r_j) < \delta$, while every other $r_{j'}$ is at distance more than δ from u. So j is the only index that passes, and $w - s_j = \mathsf{ECC.Encode}(m_j) + e$ with e of weight below δ, which $\mathsf{ECC.Decode}(\cdot)$ decodes to m_j. Base codewords and y pass no test and are handled by $\mathsf{PRC}_0.\mathsf{Decode}(\cdot)$, so correctness and soundness reduce to those of PRC_0. Finally, the event $\forall i \le q : B_\delta(c_i) \subseteq A_{\mathsf{dk}}(m_i)$ depends only on the codewords and the key. By the first part, it has the same probability for the dummies with $\mathsf{dk}^\star$ as for honest codewords with dk, which gives statistical opening robustness.

(Sketch test). By Hoeffding's inequality, as in Lemma 5, a noisy copy of r_j passes the test for r_j except with probability $e^{-\rho^2\ell/8}$, and every string independent of $r_{j'}$ passes the test for $r_{j'}$ with probability at most $e^{-\rho^2\ell/8}$. This covers the other locators, the base codewords and their noisy copies, and y. The noise on w is $\mathcal{N}_\rho(0^{n_2})$, so $\mathsf{ECC.Decode}(\cdot)$ fails with probability at most $\varepsilon_{\mathsf{ECC}}$. A union bound gives ε_{sk}, and the opened key inherits the guarantee as in the full test. □

Construction 5 meets the lower bounds for PRCs of Corollaries 1 and 4 as follows. With exact decoding, the sketch test of Proposition 3 with $\rho = 1$ (no noise), $\ell = \omega(\log\lambda)$ and ECC the identity on $\lceil\log|\mathcal{M}|\rceil$ bits gives $\kappa = Q(\log|\mathcal{M}| + \omega(\log\lambda)) + \kappa_0$, which matches Corollary 1 up to κ_0, a cost that does not grow with Q. Under $\mathcal{N}_\rho$, the locator costs $O(s/\rho^2)$ bits, which is tight up to a constant factor by Proposition 2, but the message costs n_2 bits, at least $\log|\mathcal{M}|/(1-\mathsf{h}(\frac{1-\rho}{2}))$ by the channel-coding theorem, and Proposition 4 below shows that this cost is inherent. For worst-case errors, the full test pays $n\lceil\log|\Sigma|\rceil$ bits per opening, while Corollary 4 asks only for $\log|\mathcal{M}| + \frac{2}{\ln 2}\alpha^2 n$, a gap of about $1/\alpha^2$. Replacing the stored locator by its index in a covering code (Construction 3) would close this gap for the locator, but $\mathsf{PRC}_{\mathrm{tab}}.\mathtt{KGen}(\cdot)$ and $\mathsf{NC\text{-}PRC}_{\mathrm{tab}}.\mathtt{OpenDec}(\cdot)$ would then have to round to the covering code, which is not known to be efficient, while Definition 5 requires them to be PPT. Moreover, an ECC that corrects δ worst-case errors needs $n_2 \geq 2\delta - 1$. The construction is secret-key only, since the pads s_j are part of ek.

Inherent Cost of Messages under Noise. Under noise, the message part of Construction 5 costs about $\log|\mathcal{M}|/(1-\mathsf{h}(\frac{1-\rho}{2})) \approx (2\ln 2/\rho^2)\log|\mathcal{M}|$ bits per codeword for small ρ, while a q-use code pays only $\log|\mathcal{M}|$: its key is chosen after the messages and can store each m_i directly, as in Construction 2. We show that this extra cost is inherent, and not an artifact of the one-time pad: every PRC whose key is fixed before the messages and whose codewords are perfectly uniform pays about $\log|\mathcal{M}|/\rho^2$ key bits per message (Proposition 4), so Construction 5 is optimal up to a factor $2\ln 2$. The reason is that such a key cannot store the message. The message must reach the decoder through the noisy codeword, and since the codeword is uniform, noise destroys a $1-\rho^2$ fraction of what it carries about the message and the key together. The key has to make up for this loss.

Proposition 4 (Messages Under a Key Fixed in Advance). *Let $\Sigma = \{0,1\}$ and $\rho \in (0,1]$. Let PRC be a secret-key stateful PRC with κ-bit decoder keys such that for every $\boldsymbol{m} \in \mathcal{M}^q$, the first q codewords $c_i \leftarrow \mathsf{PRC}.\mathtt{Encode}(\mathsf{ek}, m_i)$ are independent and uniform, and $\mathsf{PRC}.\mathtt{Decode}(\mathsf{dk}, c_i \oplus e) = m_i$ with probability at least $1-\varepsilon$ for $e \leftarrow\!\!\$\, \mathcal{N}_\rho(0^n)$ and each $i \in [q]$. Then*

$$\kappa \;\geq\; q \cdot \max\Big\{(1-\varepsilon)\log|\mathcal{M}| - 1,\; \Big(\frac{1-\varepsilon}{\rho^2} - 1\Big)\log|\mathcal{M}| - \frac{1}{\rho^2}\Big\}.$$

Proof. Let $\boldsymbol{m}$ be uniform on $\mathcal{M}^q$ and independent of $(\mathsf{ek}, \mathsf{dk})$, which are generated before the messages, and let $y_i := c_i \oplus e_i$ for independent $e_i \leftarrow\!\!\$\, \mathcal{N}_\rho(0^n)$. Since m_i is independent of $(\mathsf{dk}, m_{<i})$, the chain rule and Fano's inequality give

$$I(\boldsymbol{m}; \mathsf{dk}, \boldsymbol{y}) = \sum_{i\in[q]} I(m_i; \mathsf{dk}, \boldsymbol{y}, m_{<i}) \geq \sum_{i\in[q]} I(m_i; \mathsf{dk}, y_i) \geq q\big((1-\varepsilon)\log|\mathcal{M}| - 1\big),$$

because m_i is decoded from (dk, y_i) with error at most ε. The same bound holds for $I(\boldsymbol{m}; \boldsymbol{y} \mid \mathsf{dk}) = I(\boldsymbol{m}; \mathsf{dk}, \boldsymbol{y})$, as $\boldsymbol{m}$ is independent of dk.

(First term). Since $\boldsymbol{c}$ is uniform for every $\boldsymbol{m}$, the word $\boldsymbol{y}$ is uniform and independent of $\boldsymbol{m}$, so $\kappa \geq H(\mathsf{dk}) \geq I(\boldsymbol{m}; \mathsf{dk} \mid \boldsymbol{y}) = I(\boldsymbol{m}; \mathsf{dk}, \boldsymbol{y})$.

(Second term). Each bit of $\boldsymbol{y}$ is the corresponding bit of $\boldsymbol{c}$ sent through a binary symmetric channel with crossover probability $(1-\rho)/2$. For a uniform input this channel satisfies the strong data-processing inequality with constant ρ^2, and the constant tensorizes over independent uniform inputs [AG76]: for every random variable U such that $U \to \boldsymbol{c} \to \boldsymbol{y}$ is a Markov chain and $\boldsymbol{c}$ is uniform on $\{0,1\}^{qn}$, $I(U; \boldsymbol{y}) \leq \rho^2\, I(U; \boldsymbol{c})$. Take $U = (\boldsymbol{m}, \mathsf{dk})$. The chain is Markov because the noise is independent of everything else. Then

$$\begin{aligned} q\big((1-\varepsilon)\log|\mathcal{M}| - 1\big) &\leq I(\boldsymbol{m}; \boldsymbol{y} \mid \mathsf{dk}) \leq I(\boldsymbol{m}, \mathsf{dk}; \boldsymbol{y}) \leq \rho^2 I(\boldsymbol{m}, \mathsf{dk}; \boldsymbol{c}) \\ &\leq \rho^2\,(q \log|\mathcal{M}| + \kappa), \end{aligned}$$

which rearranges to the second term. □

For small ρ, a key fixed before the messages therefore pays about $\log|\mathcal{M}|/\rho^2$ bits per message, so the one-time pad of Construction 5 is optimal up to the factor $2\ln 2$ that also separates the sketch table from Proposition 2. Theorem 1 cannot see this cost, because the opened key is chosen after the messages. The proposition needs both that the key is independent of the messages and that the codewords are perfectly uniform. This covers Construction 5, even with a computationally pseudorandom PRC_0, since its first Q codewords are perfectly uniform. For a PRC whose codewords are only computationally pseudorandom, the real key has only the first property and the opened key only the second, so for such PRCs it remains open whether the message costs $\log|\mathcal{M}|/\rho^2$ key bits.

7 The Random Oracle Model

The bounds above are information-theoretic statements about decoder keys. For non-committing encryption, a programmable random oracle removes the analogous bound [Nie02]. We work in the programmable random oracle model: the oracle is a uniformly random function, and $\mathsf{NC\text{-}PRC}.\mathtt{OpenDec}(\cdot)$ may program it at points that have not been queried yet, as in the programmable oracles of [Nie02] and [CDG+18]. We here show that the answer depends on noise. We emphasize that the oracle does not help build the underlying PRC. By [GGW25, DMR25], no binary PRC robust to constant noise exists relative to a random oracle alone. The question is only whether the oracle can carry the cost of the openings. The lower bound of Section 7.2 reuses the counting lemma of Section 5.1 and the tests of Lemma 1.

7.1 RO Removes the Cost under Exact Decoding

In the plain model, the opened key must itself link each dummy to its message, and this link is what Theorem 1 counts. A programmable oracle can hold the link instead: the key only names a region of the oracle, and opening programs

the oracle so that the dummies become valid codewords under a fresh key. The simplest way to do this is to let every codeword carry a tag. A codeword is a random string u followed by the oracle's answer $H(s,u)$ under the key s, and decoding checks the tag. To open uniform dummies (u_i, w_i), the simulator picks a fresh key $s^\star$ and programs $H(s^\star, u_i) \coloneqq w_i$, one point per dummy, so the key stays λ bits however many codewords are opened.

Construction 6 (Tagged Oracle Code)**.** *Let $H : \{0,1\}^\lambda \times \{0,1\}^{n_1} \to \{0,1\}^{n_2}$ be a random oracle and $n = n_1 + n_2$. The zero-bit PRC $\mathsf{PRC}_{\mathrm{tag}} = (\mathtt{KGen}, \mathtt{Encode}, \mathtt{Decode})$ is defined as follows.*

- $\mathsf{PRC}_{\mathrm{tag}}.\mathtt{KGen}(1^\lambda)$*: sample $s \leftarrow\!\$\ \{0,1\}^\lambda$ and output $\mathsf{ek} = \mathsf{dk} = s$.*
- $\mathsf{PRC}_{\mathrm{tag}}.\mathtt{Encode}(\mathsf{ek}, m)$*: sample $u \leftarrow\!\$\ \{0,1\}^{n_1}$ and output $(u, H(s,u))$.*
- $\mathsf{PRC}_{\mathrm{tag}}.\mathtt{Decode}(\mathsf{dk}, (u,w))$*: output 1 if $w = H(s,u)$. Otherwise, output $\perp$.*

The decoder-non-committing code $\mathsf{NC\text{-}PRC}_{\mathrm{tag}} = (\mathtt{SimKGen}, \mathtt{SimCode}, \mathtt{OpenDec})$ is defined as follows.

- $\mathsf{NC\text{-}PRC}_{\mathrm{tag}}.\mathtt{SimKGen}(1^\lambda)$*: output $\mathsf{pp} \coloneqq \perp$ and $\mathsf{st} \coloneqq \perp$.*
- $\mathsf{NC\text{-}PRC}_{\mathrm{tag}}.\mathtt{SimCode}(\mathsf{pp})$*: output a uniform $c \leftarrow\!\$\ \{0,1\}^n$.*
- $\mathsf{NC\text{-}PRC}_{\mathrm{tag}}.\mathtt{OpenDec}(\mathsf{st}, \mathcal{T})$*: for $\mathcal{T} = ((m_i, (u_i, w_i)))_{i \in [q]}$, sample $s^\star \leftarrow\!\$\ \{0,1\}^\lambda$, program $H(s^\star, u_i) \coloneqq w_i$ for $i \in [q]$, and output $\mathsf{dk}^\star = s^\star$.*

Construction 6 has λ-bit keys and one oracle query, is sound with error 2^{-n_2}, and satisfies Q-bounded IND-OPEN against adversaries making T oracle queries with advantage $O((T+Q^2)/2^{\min\{\lambda, n_1\}})$. Messages are added with a second oracle, $(u, H(s,u), H'(s,u) \oplus m)$. So for exact decoding, the oracle removes the cost of Theorem 1 entirely, as for non-committing encryption. The construction is not robust to noise: an error in u changes the query. The next construction is robust, but its decoder reads every opening.

Construction 7 (Oracle Table)**.** *Let $H : \{0,1\}^\lambda \times [Q] \to \{0,1\}^n$ be a random oracle. The zero-bit stateful PRC $\mathsf{PRC}_{\mathrm{otab}} = (\mathtt{KGen}, \mathtt{Encode}, \mathtt{Decode})$ is defined as follows.*

- $\mathsf{PRC}_{\mathrm{otab}}.\mathtt{KGen}(1^\lambda)$*: sample $s \leftarrow\!\$\ \{0,1\}^\lambda$ and output $\mathsf{ek} = s$ with a counter $t \coloneqq 0$, and $\mathsf{dk} = s$.*
- $\mathsf{PRC}_{\mathrm{otab}}.\mathtt{Encode}(\mathsf{ek}, m)$*: set $t \coloneqq t+1$ and, if $t \le Q$, output $H(s,t)$.*
- $\mathsf{PRC}_{\mathrm{otab}}.\mathtt{Decode}(\mathsf{dk}, x)$*: output 1 if $\mathsf{dist}(x, H(s,j)) < \delta$ for some $j \in [Q]$. Otherwise, output $\perp$.*

The decoder-non-committing code $\mathsf{NC\text{-}PRC}_{\mathrm{otab}} = (\mathtt{SimKGen}, \mathtt{SimCode}, \mathtt{OpenDec})$ is defined as follows.

- $\mathsf{NC\text{-}PRC}_{\mathrm{otab}}.\mathtt{SimKGen}(1^\lambda)$*: output $\mathsf{pp} \coloneqq \perp$ and $\mathsf{st} \coloneqq \perp$.*
- $\mathsf{NC\text{-}PRC}_{\mathrm{otab}}.\mathtt{SimCode}(\mathsf{pp})$*: output a uniform $c \leftarrow\!\$\ \{0,1\}^n$.*
- $\mathsf{NC\text{-}PRC}_{\mathrm{otab}}.\mathtt{OpenDec}(\mathsf{st}, \mathcal{T})$*: for $\mathcal{T} = ((m_i, c_i))_{i \in [q]}$ in query order, sample $s^\star \leftarrow\!\$\ \{0,1\}^\lambda$, program $H(s^\star, i) \coloneqq c_i$ for $i \in [q]$, and output $\mathsf{dk}^\star = s^\star$.*

Proposition 5. *In the programmable random oracle model, Construction 7 is a Q-bounded zero-bit PRC with λ-bit keys and soundness $QV_\delta/2^n$. Its NC-PRC has*

uniform dummies and is δ-statistically opening-robust for every δ, and satisfies Q-bounded IND-OPEN *against adversaries making T oracle queries with advantage at most $2T/2^\lambda$. The decoder makes Q oracle queries on every input.*

Proof (Sketch). Before corruption, real codewords $H(s,1),\dots,H(s,q)$ are uniform and independent of the adversary's oracle answers unless it queries some $(s,\cdot)$, which happens with probability at most $T/2^\lambda$. In the opened game, programming fails only if the adversary queried some $(s^\star,\cdot)$, again with probability at most $T/2^\lambda$. After corruption both games consist of a uniform key s, with $H(s,i)$ equal to the i-th codeword for $i \le q$ and H uniform elsewhere. Soundness and opening robustness hold by construction. □

Both constructions need a programmable oracle. If programmed points can be detected, as for the restricted programmable random oracle of [CDG+18], they fail.

7.2 Noise Brings the Cost Back

The decoder of Construction 7 queries the programmed points on *every* input, whereas the programmed points of Construction 6 are each queried only on a negligible fraction of inputs. We show that under noise only the first kind helps. Say that a decoder is *non-adaptive* if the set $\mathcal{Q}(K,x)$ of points it queries on key K and input x does not depend on the oracle's answers, and for a key K call a point z τ-*heavy* if $w_K(z) \coloneqq \Pr_{x \leftarrow\$ \Sigma^n}[z \in \mathcal{Q}(K,x)] \ge \tau$.

Theorem 3 (Random Oracle, Non-Adaptive Decoders). *Let $\Sigma = \{0,1\}$ and let $\rho \in (0,1)$ be a constant. In the programmable random oracle model, let* PRC *and* NC-PRC *satisfy the hypotheses of Lemma 1, where the decoder is non-adaptive and makes at most $T \ge 1$ queries with ℓ_{out}-bit answers, and* NC-PRC.$\mathtt{OpenDec}(\cdot)$ *programs at most $P \ge 1$ points. Let $p \ge 2$ with $p = \mathrm{poly}(\lambda)$, let $\theta \in (0,1)$, and let $\tau \in (0,1]$ satisfy*

$$\tau \le \frac{1}{pP} \qquad \text{and} \qquad 64\,P^2\,T\,\tau^{(1-\rho^2)/(1+\rho^2)} \le (16\,p\,|\mathcal{M}|)^{-1/\theta}. \tag{1}$$

Fix $q \le Q$ and suppose that, for uniform messages $\boldsymbol{m} \leftarrow\$ \mathcal{M}^q$, with probability at least $1/2$ at most h of the programmed points are τ-heavy for the opened key. Then, for all sufficiently large λ,

$$\kappa + h\left(\ell_{\mathrm{out}} + \log(T/\tau)\right) + \log(P+1) \;\ge\; (1-\theta)\,q\left(\log|\mathcal{M}| + \log p - c_\theta\right),$$

where $c_\theta \coloneqq 2 + 3/(1-\theta)$. If p, P, T and $|\mathcal{M}|$ are $\mathrm{poly}(\lambda)$ and θ is constant, then Eq. (1) holds for some $\tau = 1/\mathrm{poly}(\lambda)$.

Proof. If $\kappa \ge q(\log|\mathcal{M}| + \log p)$ there is nothing to prove, so assume $\kappa < q(\log|\mathcal{M}| + \log p)$. Since at most P points are programmed, we may also assume $h \le P$. Let H_0 be the oracle before programming and H_1 the oracle after

$\mathsf{NC\text{-}PRC.OpenDec}(\cdot)$ has programmed it. The dummies and the messages are uniform and independent of H_0. Let $\gamma := 1/(8P)$ and $\eta := 64P^2T\,\tau^{(1-\rho^2)/(1+\rho^2)}$.

(Light points are rarely queried under noise). Fix a key K and let $f_z(x) := \mathbf{1}[z \in \mathcal{Q}(K,x)]$. Since the decoder is non-adaptive, f_z depends only on K and not on the oracle, $\mathbb{E}_x[f_z(x)] = w_K(z)$, and $\sum_z w_K(z) \le T$. For a word c, the value $T_\rho f_z(c) = \Pr_e[z \in \mathcal{Q}(K, c \oplus e)]$ with $e \leftarrow\!\$\, \mathcal{N}_\rho(0^n)$ is the probability that z is queried on a noisy copy of c. By hypercontractivity, $\|T_\rho f_z\|_2 \le \|f_z\|_{1+\rho^2} = w_K(z)^{1/(1+\rho^2)}$, so Markov's inequality gives $\mu\{c : T_\rho f_z(c) \ge \gamma\} \le \gamma^{-2} w_K(z)^{2/(1+\rho^2)}$. Call c *bad for* K if $T_\rho f_z(c) \ge \gamma$ for some point z that is not τ-heavy for K, and *good for* K otherwise. Since $w_K(z)^{2/(1+\rho^2)} \le \tau^{(1-\rho^2)/(1+\rho^2)}\, w_K(z)$ for such points, the bad words have measure at most

$$\gamma^{-2}\,\tau^{(1-\rho^2)/(1+\rho^2)} \sum_z w_K(z) \le \eta.$$

(Few dummies are bad for the opened key). The opened key depends on the dummies, so we take a union bound over all keys. For a fixed K, the dummies are independent and each is bad for K with probability at most η, so more than θq of them are bad with probability at most $2^q \eta^{\theta q}$. By Eq. (1) and the assumption on κ,

$$\begin{aligned}\Pr\big[\exists K : \text{more than } \theta q \text{ dummies are bad for } K\big] &\le 2^{\kappa+q}\,\eta^{\theta q} \\ &< (2p\,|\mathcal{M}|)^q\,(16\,p\,|\mathcal{M}|)^{-q} \le \frac{1}{8}.\end{aligned}$$

(An oracle-free key). For a key K and a list A of at most h points that are τ-heavy for K, each with an ℓ_{out}-bit value, let $\mathsf{Decode}'(K, A, x)$ run $\mathsf{PRC.Decode}(K, x)$ and answer the points of A with the listed values and all other points with H_0. For a fixed H_0 this is a fixed function of (K, A, x). Since $\sum_z w_K(z) \le T$, at most T/τ points are τ-heavy for K, so the number of pairs (K, A) is at most

$$N := 2^\kappa\,(P+1)\,\big(2^{\ell_{\text{out}}}\,T/\tau\big)^h.$$

Let G be the event that (a) at most h programmed points are τ-heavy for $\mathsf{dk}^\star$, (b) $\mathsf{dk}^\star$ accepts at most a $1/p$ fraction of all strings under H_1, (c) for every i, a noisy copy of c_i decodes to m_i under $\mathsf{dk}^\star$ and H_1 with probability at least $3/4$ over the noise, and (d) at most θq dummies are bad for $\mathsf{dk}^\star$. The tests of Lemma 1 only run the decoder and therefore carry over to the oracle model, where they are passed relative to H_1. Hence (b) fails with negligible probability by Lemma 1(3), and (c) fails with negligible probability by Lemma 1(2) and Markov's inequality. With the hypothesis on (a) and the bound above for (d), $\Pr[G] \ge 1/2 - 1/8 - \mathsf{negl}(\lambda) \ge 1/4$ for large λ.

Assume G, and let A list the programmed points that are τ-heavy for $\mathsf{dk}^\star$, with their programmed values. Then $\mathsf{Decode}'(\mathsf{dk}^\star, A, x)$ can differ from the real decoder under H_1 only if $\mathcal{Q}(\mathsf{dk}^\star, x)$ contains a programmed point that is not

τ-heavy. For uniform x this happens with probability at most $P\tau \le 1/p$, so $\mathtt{Decode}'(\mathtt{dk}^\star, A, \cdot)$ accepts at most a $2/p$ fraction of all strings. For a noisy copy of a good dummy it happens with probability at most $P\gamma = 1/8$, so by (c) the noisy copy decodes to m_i under $\mathtt{Decode}'(\mathtt{dk}^\star, A, \cdot)$ with probability at least $3/4 - 1/8 \ge 1/2$.

(Counting). Fix H_0, and consider a pair (K, A) for which $\mathtt{Decode}'(K, A, \cdot)$ accepts at most a $2/p$ fraction of all strings. As in the noisy case of Theorem 1, a uniform pair (c, m) satisfies $\Pr_e[\mathtt{Decode}'(K, A, c \oplus e) = m] \ge 1/2$ with probability at most $\beta := 4/(p\,|\mathcal{M}|)$. Under G, the pair $(\mathtt{dk}^\star, A)$ is of this kind and satisfies this condition for all good dummies, of which there are at least $(1-\theta)q$. A union bound over the pairs (K, A) and over the at most 2^q sets of good dummies gives

$$\frac{1}{4} \le \Pr[G] \le N \cdot 2^q \cdot \beta^{(1-\theta)q}.$$

Taking logarithms and using $q + 2 \le 3q$,

$$\log N \;\ge\; (1-\theta)\,q\,(\log|\mathcal{M}| + \log p - 2) - q - 2 \;\ge\; (1-\theta)\,q\,\big(\log|\mathcal{M}| + \log p - c_\theta\big),$$

which is the claim, since $\log N = \kappa + h\,(\ell_{\mathrm{out}} + \log(T/\tau)) + \log(P+1)$.

Finally, for constant $\rho < 1$ the exponent $(1-\rho^2)/(1+\rho^2)$ is a positive constant, so if p, P, T and $|\mathcal{M}|$ are $\mathrm{poly}(\lambda)$ and θ is constant, both conditions of Eq. (1) hold for a sufficiently small $\tau = 1/\mathrm{poly}(\lambda)$. □

So each opening is paid for either in key bits or in programmed oracle points that the decoder reads on a noticeable fraction of inputs. Since $\sum_z w_K(z) \le T$, at most T/τ points are heavy, so $h \le T/\tau$ always holds, which gives the form $\kappa + \tilde{O}(T\ell_{\mathrm{out}}/\tau) \ge \Omega(q(\log|\mathcal{M}| + \log p))$. The light programmed points of Construction 6, which suffice for non-committing encryption, are useless under noise. In Construction 7, in contrast, all q programmed points have weight 1, and the theorem only says that $\lambda + q\,(n + \log(Q/\tau))$ exceeds the right-hand side. The threshold τ is $1/\mathrm{poly}(\lambda)$ only if $|\mathcal{M}| = \mathrm{poly}(\lambda)$, which includes zero-bit codes. For longer messages Eq. (1) forces a smaller τ. The proof is a variant of the compiler of [GGW25]. Its main obstacle, that codewords must stay close to uniform given the key, is replaced by a union bound over keys, which is possible because the dummies are uniform and, for non-adaptive decoders, the split into heavy and light points depends only on the key.

Remark 4 (Adaptive decoders). We do not know whether Theorem 3 holds for adaptive decoders. In a multi-bit variant of Construction 7, the decoder can query a pad $H'(s, j^\star)$ only after locating $j^\star$. This query is light but made on every noisy copy of the $j^\star$-th codeword. So for adaptive decoders, the message part may be carried by light queries, and only the locating part should be expensive.

Generative AI Disclaimer. Claude Fable 5.1 and Opus 5.5 were used throughout the preparation of this paper for discussion and brainstorming, and for drafting and revising prose. The technical contributions, including the ideas,

definitions, protocols, and proofs, are the authors' own, and all AI-assisted text, code, and measurements were reviewed by the authors, who take full responsibility for the content of this paper.

References

AAC+25. Omar Alrabiah, Prabhanjan Ananth, Miranda Christ, Yevgeniy Dodis, and Sam Gunn. Ideal pseudorandom codes. In Michal Koucký and Nikhil Bansal, editors, *57th ACM STOC*, pages 1638–1647. ACM Press, June 2025. doi:10.1145/3717823.3718309.

AG76. Rudolf Ahlswede and Péter Gács. Spreading of sets in product spaces and hypercontraction of the Markov operator. *The Annals of Probability*, 4(6):925–939, 1976.

Bec75. William Beckner. Inequalities in Fourier analysis. *Annals of Mathematics*, 102(1):159–182, 1975.

Bon70. Aline Bonami. Étude des coefficients de Fourier des fonctions de $L^p(G)$. *Annales de l'Institut Fourier*, 20(2):335–402, 1970.

Cao26. Ganyuan Cao. Ideal pseudorandom code, revisited. CANS 2026. Cryptology ePrint Archive, Paper 2026/1657, 2026. URL: https://eprint.iacr.org/2026/1657.

CDG+18. Jan Camenisch, Manu Drijvers, Tommaso Gagliardoni, Anja Lehmann, and Gregory Neven. The wonderful world of global random oracles. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 280–312. Springer, Cham, April / May 2018. doi:10.1007/978-3-319-78381-9_11.

CFGN96. Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996. doi:10.1145/237814.238015.

CG24. Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VI*, volume 14925 of *LNCS*, pages 325–347. Springer, Cham, August 2024. doi:10.1007/978-3-031-68391-6_10.

CGG+25. Miranda Christ, Noah Golowich, Sam Gunn, Ankur Moitra, and Daniel Wichs. Improved pseudorandom codes from permuted puzzles. Cryptology ePrint Archive, Report 2025/2222, 2025. URL: https://eprint.iacr.org/2025/2222.

CHK05. Ran Canetti, Shai Halevi, and Jonathan Katz. Adaptively-secure, non-interactive public-key encryption. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 150–168. Springer, Berlin, Heidelberg, February 2005. doi:10.1007/978-3-540-30576-7_9.

CHLL97. Gérard Cohen, Iiro Honkala, Simon Litsyn, and Antoine Lobstein. *Covering Codes*, volume 54 of *North-Holland Mathematical Library*. Elsevier, 1997.

CMY26. Yu Chen, Xinyu Mao, and Hongxu Yi. Advancing pseudorandom codes: Beyond parity checks and standard-model CCA1 security. Cryptology ePrint Archive, Paper 2026/1228, 2026. URL: https://eprint.iacr.org/2026/1228.

DFV26. Victor Dyseryn, Danilo Francati, and Daniele Venturi. Public-key pseudorandom codes from distorted mceliece assumptions. Cryptology ePrint Archive, Paper 2026/1184, 2026. URL: https://eprint.iacr.org/2026/1184.

DJK+26. Nico Döttling, Antoine Joux, Venkata Koppula, Mahesh Sreekumar Rajasree, and Hendrik Waldner. Chosen ciphertext secure pseudorandom codes in the standard model. In *CRYPTO*, 2026. URL: https://eprint.iacr.org/2026/1227.

DKRR26. Nico Döttling, Venkata Koppula, Mahesh Sreekumar Rajasree, and Felix Rohrbach. On robust combiners for pseudorandom codes and PKE with pseudorandom ciphertexts. TCC, 2026. URL: https://eprint.iacr.org/2026/2650.

DMR25. Nico Döttling, Anne Müller, and Mahesh Sreekumar Rajasree. Separating pseudorandom codes from local oracles. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part IV*, volume 16271 of *LNCS*, pages 225–257. Springer, Cham, December 2025. doi:10.1007/978-3-032-12290-2_8.

DN00. Ivan Damgård and Jesper Buus Nielsen. Improved non-committing encryption schemes based on a general complexity assumption. In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 432–450. Springer, Berlin, Heidelberg, August 2000. doi:10.1007/3-540-44598-6_27.

FGP+26. Danilo Francati, Yevin Nikhel Goonatilake, Shubham Pawar, Daniele Venturi, and Giuseppe Ateniese. The coding limits of robust watermarking for generative models. In *IEEE European Symposium on Security and Privacy (EuroS&P)*, 2026. URL: https://eprint.iacr.org/2025/1620.

GG24. Surendra Ghentiyala and Venkatesan Guruswami. New constructions of pseudorandom codes. Cryptology ePrint Archive, Report 2024/1425, 2024. URL: https://eprint.iacr.org/2024/1425.

GGW25. Sanjam Garg, Sam Gunn, and Mingyuan Wang. Black-box crypto is useless for pseudorandom codes. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part IV*, volume 16271 of *LNCS*, pages 205–224. Springer, Cham, December 2025. doi:10.1007/978-3-032-12290-2_7.

GM24. Noah Golowich and Ankur Moitra. Edit distance robust watermarks via indexing pseudorandom codes. *Advances in Neural Information Processing Systems*, 37:20645–20693, 2024.

GZS24. Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. Cryptology ePrint Archive, Report 2024/1597, 2024. URL: https://eprint.iacr.org/2024/1597.

Har66. Lawrence H. Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1(3):385–393, 1966.

LSS26. Huijia Lin, Kameron Shahabi, and Min Jae Song. Unforgeable watermarks for language models via robust signatures. In *CRYPTO*, 2026. URL: https://eprint.iacr.org/2026/282.

McD89. Colin McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics*, volume 141 of *London Mathematical Society Lecture Note Series*, pages 148–188. Cambridge University Press, 1989.

Nie02. Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 111–126. Springer, Berlin, Heidelberg, August 2002. doi:10.1007/3-540-45708-9_8.

O'D14. Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

A Related Work

Pseudorandom Codes. PRCs were introduced in [CG24] from LDPC codes and LPN-type assumptions, and applied to watermarking language models and images [CG24, GZS24] and to edit-distance robust watermarks [GM24]. Alrabiah et al. [AAC+25] defined ideal and CCA security, and later works [DJK+26, CMY26] gave CCA-secure constructions in the standard model. Lin et al. [LSS26] use PRCs inside unforgeable watermarks. Garg, Gunn and Wang [GGW25] and Döttling, Müller and Rajasree [DMR25] show that no binary PRC robust to constant noise exists relative to a random oracle or local oracles alone. Both use a counting or entropy argument over oracle-free keys, which we compare with ours in Section 5.2. Döttling et al. [DKRR26] rule out black-box robust combiners and introduce a random PRC model, and Francati et al. [FGP+26] bound the robustness of any such code by $1 - 1/|\Sigma|$.

Non-Committing Encryption. Non-committing encryption was introduced by Canetti et al. [CFGN96] for adaptive security of multi-party computation, with later constructions in [DN00, CHK05]. Nielsen [Nie02] showed that non-interactive non-committing encryption needs keys as long as all messages it opens, and that a programmable random oracle removes this bound. Our lower bounds contain Nielsen's bound as the message term, and our random-oracle results show that, unlike for encryption, the oracle does not help once codewords must survive noise. If programmed points can be detected, as for the restricted programmable random oracle of [CDG+18], even the noise-free constructions fail.

Adaptive Security for PRCs. Our prior work [Cao26] gives a UC treatment of PRCs, derives NC-PRCs from adaptive UC security, introduces targeted low opening capacity (TLOC), and shows that LDPC-based PRCs have it. Here we work only with games, and Corollary 2 shows that a variant of TLOC over sound keys holds for every PRC once the number of openings exceeds $(\kappa + \lambda)/(\log |\mathcal{M}| + \log p)$.