Detection of Algebraic Manipulation in the Presence of Leakage

Hadi Ahmadi and Reihaneh Safavi-Naini

Department of Computer Science, University of Calgary
{hahmadi, rei}@ucalgary.ca

Abstract. We investigate the problem of algebraic manipulation detection (AMD) over a communication channel that partially leaks information to an adversary. We assume the adversary is computationally unbounded and there is no shared key or correlated randomness between the sender and the receiver. We introduce leakage-resilient (LR)-AMD codes to detect algebraic manipulation in this model.

We consider two leakage models. The first model, called *linear leakage*, requires the adversary's uncertainty (entropy) about the message (or encoding randomness) to be a constant fraction of its length. This model can be seen as an extension of the original AMD study by Cramer et al. [2] to when some leakage to the adversary is allowed. We study *randomized strong* and *deterministic weak* constructions of linear (L)LR-AMD codes. We derive lower and upper bounds on the redundancy of these codes and show that known optimal (in rate) AMD code constructions can serve as optimal LLR-AMD codes. In the second model, called *block leakage*, the message consists of a sequence of blocks and at least one block remains with uncertainty that is a constant fraction of the block length. We focus on deterministic block (B)LR-AMD codes. We observe that designing optimal such codes is more challenging: LLR-AMD constructions cannot function optimally under block leakage. We thus introduce a new optimal BLR-AMD code construction and prove its security in the model.

We show an application of LR-AMD codes to tampering detection over wiretap channels. We next show how to compose our BLR-AMD construction, with a few other keyless primitives, to provide both integrity and confidentiality in transmission of messages/keys over such channels. This is the best known solution in terms of randomness and code redundancy. We discuss our results and suggest directions for future research.

1 Introduction

In a basic message authentication scenario, Alice wants to deliver a message to Bob in the presence of Eve, who can arbitrarily manipulate the communication. The goal is to enable Bob to detect adversarial manipulation with high probability. This objective is achieved by appending to the message a relatively short authentication tag, calculated based on the message and a shared secret key between the legitimate parties. In the computational setting, message authentication is also attained via public key cryptography using digital signatures. The classical message authentication problem adopts the strong Dolev-Yao attacker model [4], which possesses complete read and write access to the communication and modifies messages arbitrarily in real-time. Keyless detection of such a powerful adversarial manipulation is impossible. When a less powerful adversary is present however, alternative solutions to keyless manipulation detection may exist. In this work, we consider a theoretical model of communication where Alice is connected to Bob through a channel whose content can be manipulated by an additive (algebraic) noise chosen by Eve. There is no shared key between Alice and Bob and the adversary is computationally unbounded.

Detection of algebraic manipulation has already been studied by Cramer et al. [2]. There, the authors assumed that the communication system keeps its content "private" and designed *algebraic manipulation detection* (AMD) codes to provide message integrity, only when the adversary cannot view the codeword. This restrictive assumption, however, makes the adversary of an oblivious nature since manipulation will be solely based on the public codebook knowledge. We relax this assumption and study leakage-resilient (LR)-AMD codes for situations where the adversary obtains partial information about the codeword.

1.1 Problem definition and results

An *LR-AMD* code is defined by a pair of encoding and decoding functions. When a message is encoded, the codeword is "partially" leaked to Eve. She then uses this to determine an arbitrary noise variable and adds it to the codeword. We say that decoding fails if the manipulated codeword is decoded to a message other than the original one. The LR-AMD code must satisfy *correctness* and *security.* Correctness means in the absence of noise, decoder returns the original message. Security means small decoding failure probability for a non-zero adversarial noise. The *optimality* of a code construction on the other hand is measured via *effective tag length* or *asymptotic rate*: The former is the code redundancy and the latter is the asymptotic message length divided by the code length.

We define two classes of LR-AMD codes, namely *linear* (L)LR-AMD and *block* (B)LR-AMD codes. LLR-AMD coding is an extension of AMD coding [2] to when Eve's uncertainty about the message (or code randomness) stays proportional to the length. We consider *deterministic weak LLR-AMD codes* which provide security guarantee for a randomly chosen message as well as *randomized strong LLR-AMD codes* that provide security for any message. BLR-AMD codes are for detecting algebraic manipulation in the block leakage scenario, where the message is a sequence of blocks and Eve's uncertainty for (at least) one block stays proportional to its length. We only focus on *deterministic weak* BLR-AMD codes. The leakage in LR-AMD codes is specified by *leakage rate* $0 \leq \alpha \leq 1$, i.e., the fraction of message/randomness that can be leaked in terms of min-entropy.

AMD codes vs. LLR-AMD codes. We show that optimal AMD code constructions work optimally as well under linear leakage. We first prove general bounds on the failure probability of AMD codes when used in the linear leakage model. Applying these results to optimal AMD constructions suggests strong

LLR-AMD constructions with the asymptotic rate of 1 and weak LR-AMD codes with the asymptotic rate of $1/(1+\alpha)$. This implies upper bounds on the effective tag lengths of weak and strong LLR-AMD code families. The more challenging question is whether the bounds can be improved, especially for weak codes. The answer is negative: We derive lower bound expressions on the effective tag lengths of LLR-AMD code constructions, which are (almost) equal to the upper bounds, thus implying the optimality of the code constructions.

BLR-AMD codes. It is impossible to accomplish deterministic LLR-AMD with rate over $1/(1+\alpha)$, revealing that when α tends to 1 the maximum achievable rate is bounded by 1/2. This leads us to a question whether there are reasonably interesting leakage scenarios for which deterministic AMD with higher rates (less redundancy) is possible. We consider the *block-leakage* model, described above, and introduce an efficient systematic BLR-AMD code construction that that achieves the asymptotic rate of 1. We note that this construction can be used as a weak LLR-AMD code and also a strong LLR-AMD code by choosing part of the message string to be used for encoding randomness.

Manipulation detection over wiretap channels. In the wiretap channel [12], the sender sends a message to the receiver over the main channel, while the eavesdropper receives a noisy version via a probabilistic wiretapping channel. Wyner showed that transmission with perfect security is possible using randomized wiretap codes [12]. To protect against tampering however, one needs keyless manipulation detection which is impossible if the adversary's manipulation power is unrestricted. We thus restrict the adversary to "algebraic manipulation" over the wiretap channel. We consider a wiretap channel with noise-free main channel and u-ary erasure/symmetric wiretapping component with symbol erasure/corruption probability p. We show that the LLR-AMD codes detect algebraic manipulation when $p > 0.5$, whereas the BLR-AMD code construction protects against a wider range of p. Finally we consider the case that symbols are binary and manipulation is general. We will use the following construction. Alice encodes her message using a BLR-AMD code, passes it to a Manchester encoder, and transmits the resulting codeword over the channel bit-by-bit via on-off keying. We will argue that the combination of Manchester coding and on-off keying restrict the manipulation of the adversary to algebraic ones, which can be detected with high probability by our BLR-AMD code construction. The above construction can be composed with wiretap codes to provide both privacy and manipulation detection in secret key/message transport.

1.2 Discussion and related work

Error correcting codes. Shannon's seminal work [11] provides the first formal treatment of *reliable message transmission* when the communication channel is corrupted by *probabilistic noise*. The adversarial channel model was later proposed by Hamming [9] as an alternative to Shannon's model. Existence and construction of error correcting codes over oblivious adversarial channels (corrupting up to a p-fraction of bits) has been studied in [8, 10]. Our goal in this paper is *detection of errors* in adversarial channels.

Deterministic vs. randomized coding. We study both randomized and deterministic LR-AMD codes. Randomized coding is interesting as it allows us to detect algebraic manipulation of any messages, as opposed to a random message. But nevertheless, the study of deterministic code constructions is crucial because generating "true" randomness can be hard, e.g., for low-cost devices. When true randomness is not available but the input message itself is a (random) secret key deterministic LR-AMD coding becomes interesting.

Communication channel model. The application of LR-AMD codes to tampering detection over wiretap channels suites for instance a scenario where a covert adversary tries not to use high-energy jamming/overshadowing attacks to avoid the risk of being detected. This adversary rathers annihilate, amplify, and/or flip communication symbols using same energy signals. When binary modulation is used, this is translated as the four bitwise tampering functions: keep, flip, set-to-0, and set-to-1. Binary modulation is popular in many communication systems such as fiber optics.

Integrity codes. We show an interesting application the BLR-AMD codes for message integrity over tamperable wiretap channels. Similar problem has been addressed by integrity codes [1]. We mention the main advantages of our approach over the solution in [1]. The construction of an integrity code consists of on-off keying and unidirectional coding. The authors realize that on-off keying does not prevent all 1-to-0 errors if the adversary knows the modulator carrier. They resolve this by encoding bit "1" to a long random (e.g., 48-bit [1, Section 4]) string. This solution however requires a lot of local secret randomness (per transmitted bit) and causes a huge bandwidth waste by drastically decreasing the transmission rate. Our approach alternatively benefits from the BLR-AMD code construction that detects 1-to-0 conversions made by bit-flipping: It does not need randomness and more importantly is much more efficient in rate.

Non-malleable codes. Dziembowski el al. [6] introduced non-malleable (NM) codes which relax the definition of error correction and detection: non-malleability requires manipulation to result either in the original message or in an unrelated variable. NM codes have found application in algorithmic tamper-proof security [7]. Authors of [6] built an NM code construction for bitwise manipulation which takes advantage of AMD codes. This sparks the idea of using LR-AMD codes to build NM codes for leakage scenarios.

2 Notations and Preliminaries

We use calligraphic $\mathcal{X}$ and bold $\mathbf{X}$ letters to denote sets and their sizes, and use uppercase X and lowercase x letters to denote random variables and their realizations over sets. X^n indicates a sequence of length n and X_i represents its i-th element. We use $\Pr_X(\mathcal{E})$ to show the probability of $\mathcal{E}$ over distribution X, and use $E_x(Y)$ to indicate the expectation of Y over choices of x. Logarithms are by default to base 2. The following definitions are used throughout the paper.

Definition 1 (Min-entropy). *For a random variable $X \in \mathcal{X}$ with distribution P_X, its min-entropy is obtained as $H_\infty(X) = -\log \max_x P_X(x)$.*

Definition 2 (Conditional min-entropy). *Given random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ with joint distribution P_{XY}, the (average) conditional min-entropy of X given Y is obtained as $\tilde{H}_\infty(X|Y) = -\log(E_y\left(\max_x P_{X|Y}(x|y)\right))$.*

Definition 3 (Weak source). *A random variable X over the set $\mathcal{X}$ of size $\mathbf{X}$ is called a* β-weak source *if it holds $H_\infty(X) \geq \beta \log \mathbf{X}$. The source is called* β-weak conditioned on the random variable Z *if it holds $\tilde{H}_\infty(X|Z) \geq \beta \log \mathbf{X}$.*

3 LR-AMD Codes: Definitions

A leakage-resilient algebraic manipulation detection (LR-AMD) code is specified by a pair of encoding/decoding functions $Enc : \mathcal{M} \rightarrow \mathcal{X}$ and $Dec : \mathcal{X} \rightarrow \mathcal{M} \cup \{\perp\}$, where $\mathcal{M}$ is the message space, $\mathcal{X}$ is the additive group of the codeword space, and $\perp$ is the manipulation detection symbol. Figure 1 illustrates Alice using this code to send Bob a message M over an algebraically manipulable channel with leakage. Alice encodes $X = Enc(M)$ and sends it. The channel leaks information Z to Eve. Eve uses Z to choose $\Delta \in \mathcal{X}$ and replaces X with $Y = X + \Delta$. Bob receives Y and decodes it to $\hat{M} = Dec(Y)$. We say *decoding fails* if $\hat{M} \notin \{M, \perp\}$.

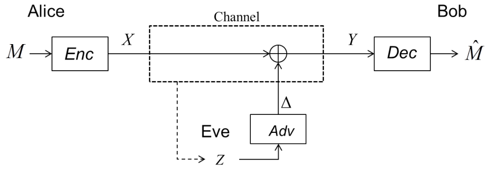


Fig. 1. Algebraic manipulation with leakage.

An LR-AMD code must satisfy *correctness* and *security*: The former means decoding of encoding of a message should return the message itself, and the latter requires negligible failure probability (when $\Delta \neq 0$). Depending on whether security is for a random message or for all messages, we define weak and strong LR-AMD codes, respectively. The random-message security for a weak LR-AMD code lets the encoding function be deterministic. In this work, we only consider "deterministic" weak LR-AMD codes. A strong LR-AMD code, however, must be randomized to work for all messages. We define two classes of LR-AMD, namely LLR-AMD and BLR-AMD, codes. Throughout, we let $0 \leq \alpha, \epsilon \leq 1$ be real values and $\mathcal{M}$, $\mathcal{R}$, and $\mathcal{X}$ be the message, randomness (if applicable), and codeword spaces of sizes $\mathbf{M} = |\mathcal{M}|$, $\mathbf{R} = |\mathcal{R}|$, and $\mathbf{X} = |\mathcal{X}|$, respectively.

3.1 LLR-AMD codes

A linear (L)LR-AMD code guarantees security if the message/randomness min-entropy is above a certain fraction of its length given the leakage information.

Definition 4 (Weak LLR-AMD code). *The deterministic block code with encoding function* $Enc : \mathcal{M} \rightarrow \mathcal{X}$ *and decoding function* $Dec : \mathcal{X} \rightarrow \mathcal{M} \cup \{\perp\}$ *is a* $(\mathbf{M}, \mathbf{X}, \alpha, \epsilon)$*-weak LLR-AMD code if* $\forall m : Dec(Enc(m)) = m$*, and for any adversary* $\mathcal{A}dv$ *and variables* $M \in \mathcal{M}$ *and* Z *such that* M *is* $(1-\alpha)$*-weak conditioned on* Z*, it holds:*

$$\Pr_{M,\mathcal{A}dv}\Big(Dec(Enc(M) + \mathcal{A}dv(Z)) \notin \{M, \perp\}\Big) \leq \epsilon. \tag{1}$$

The code is systematic if $Enc(M) = (M, \mathbf{Tag}(M))$ *for* $\mathbf{Tag} : \mathcal{M} \rightarrow \mathcal{T}$*, where* $\mathcal{M}$ *and* $\mathcal{T}$ *are additive groups.*

Definition 5 (Strong LLR-AMD Code). *The randomized block code with encoding function* $Enc : \mathcal{R} \times \mathcal{M} \rightarrow \mathcal{X}$ *and decoding function* $Dec : \mathcal{X} \rightarrow \mathcal{M} \cup \{\perp\}$ *is a* $(\mathbf{M}, \mathbf{X}, \mathbf{R}, \alpha, \epsilon)$*-strong LLR-AMD code if* $\forall m : Dec(Enc(m)) = m$*, and for any adversary* $\mathcal{A}dv$ *and variables* $R \in \mathcal{R}$ *and* Z *such that* R *is* $(1-\alpha)$*-weak conditioned on* Z*,*

$$\forall m : \quad \Pr_{R,\mathcal{A}dv}\Big(Dec(Enc(R; m) + \mathcal{A}dv(Z)) \notin \{m, \perp\}\Big) \leq \epsilon. \tag{2}$$

The code is systematic if $Enc(R; M) = (M, \mathbf{Tag}(R; M))$ *for some function* $\mathbf{Tag} : \mathcal{R} \times \mathcal{M} \rightarrow \mathcal{R} \times \mathcal{G}$*, where* $\mathcal{M}$*,* $\mathcal{R}$*, and* $\mathcal{G}$ *are additive groups.*

Remark 1. Definitions 4 and 5 restrict leakage in terms of leftover min-entropy. This is a general form of that used by the leakage-resilient cryptography literature [5] which assumes leakage of a uniform source via a limited-length function.

For consistency with [2] when there is no leakage ($\alpha = 0$), we drop α from the notation and use $(\mathbf{M}, \mathbf{X}, \epsilon)$-weak AMD and $(\mathbf{M}, \mathbf{X}, \mathbf{R}, \epsilon)$-strong AMD codes.

3.2 BLR-AMD codes

The block leakage model captures a scenario where the message is a sequence of (equal-sized) blocks and the leakage information leaves (at least) one message block with some leftover min-entropy proportional to its length. A BLR-AMD code is a scheme that detects algebraic manipulation with the codeword in the block leakage model. Here, we focus on deterministic weak BLR-AMD codes.

Definition 6 (BLR-AMD code). *Let* $Enc : \mathcal{U}^d \rightarrow \mathcal{X}$ *and* $Dec : \mathcal{X} \rightarrow \mathcal{U}^d \cup \{\perp\}$ *denote a deterministic block code. For* $\mathbf{U} = |\mathcal{U}|$, $\mathbf{X} = |\mathcal{X}|$, $0 \leq \alpha < 1$ *and* $0 < \epsilon \leq 1$*, the code is a* $(\mathbf{U}^d, \mathbf{X}, \alpha, \epsilon)$*-(weak)BLR-AMD code if for any adversary* $\mathcal{A}dv$*, message* $M \in \mathcal{U}^d$ *and leakage* Z *such that* $\exists o \in \{1, \ldots, d\} : \tilde{H}_\infty(M_o | Z, (M_j)_{j \neq o}) \geq (1-\alpha) \log \mathbf{U}$*, the security property (1) holds.*

An instance of block leakage is when the message is a uniform secret and the adversary can observe $Z = (f_1(M_1), \ldots, f_d(M_d))$, for d arbitrary functions f_1 to f_d, provided that the sum of function lengths stays $\leq \alpha d \log \mathbf{U}$. This follows that at least one of the functions f_o should be of length $\leq \alpha \log \mathbf{U}$, satisfying the block leakage model. Another scenario where BLR-AMD codes can be used is the tamperable wiretap channel, discussed in Section 5.

3.3 LR-AMD code optimality

It is of theoretical and practical significance to design LR-AMD code constructions with flexible parameters, rather than a single code.

Definition 7 (LR-AMD code family). *A class $\mathcal{F}$ of LR-AMD codes is called an LR-AMD code family if for any integers $\kappa, \nu \in \mathbb{N}$ and real $0 \leq \alpha \leq 1$, it contains an LR-AMD code with message size $\mathbf{M} \geq 2^{\nu}$ and failure probability $\epsilon \leq 2^{-\kappa}$ for leakage rate α.*

We use *effective tag length* [1] and *asymptotic code rate* to measure the optimality of an LR-AMD code family in concrete and asymptotic ways, respectively.

Definition 8 (Effective tag length). *For $\kappa, \nu \in \mathbb{N}$, $0 \leq \alpha \leq 1$, the effective tag length of an LR-AMD code family $\mathcal{F}$ is $\varpi^*_{\mathcal{F}}(\kappa, \nu, \alpha) = \min_{\mathcal{F}^*} \log \mathbf{X} - \nu$ where $\mathcal{F}^* \subseteq \mathcal{F}$ has all codes with $\mathbf{M} \geq 2^{\nu}$ and $\epsilon \leq 2^{-\kappa}$ for leakage rate α.*

Definition 9 (Asymptotic rate). *For $0 \leq \alpha \leq 1$, the asymptotic rate of an LR-AMD code family $\mathcal{F}$ equals $Rate_{\mathcal{F}}(\alpha) = \lim_{\kappa \to \infty} \max_{\nu} \max_{\mathcal{F}^*} \frac{\nu}{\log \mathbf{X}}$ where $\mathcal{F}^* \subseteq \mathcal{F}$ has all codes with $\mathbf{M} \geq 2^{\nu}$ and $\epsilon \leq 2^{-\kappa}$ for leakage rate α.*

4 Optimal LR-AMD Constructions

4.1 LLR-AMD code constructions

This section aims to give optimal and efficient constructions of weak and strong LLR-AMD code families. We show that there is no need for designing new codes since an optimal AMD code construction (for no leakage) works almost optimally when there is linear leakage. We show this by (1) proving general upper-bounds on the failure probability of weak and strong AMD codes when used under linear leakage, and (2) proving lower-bounds on the effective tag length (and failure probability) of LLR-AMD code families. The former is shown below.

Theorem 1 (Appendix A). *Any $(\mathbf{M}, \mathbf{X}, \mathbf{R}, \epsilon)$-strong AMD code is a $(\mathbf{M}, \mathbf{X}, \mathbf{R}, \alpha, \mathbf{R}^{\alpha}\epsilon)$-strong LLR-AMD code, and any $(\mathbf{M}, \mathbf{X}, \epsilon)$-weak AMD code is a $(\mathbf{M}, \mathbf{X}, \alpha, \mathbf{M}^{\alpha}\epsilon)$-weak LLR-AMD code.*

We apply the above result to examples of optimal AMD code constructions. Lemma 1 shows a strong AMD construction suggested by Cramer et al. [2].

Lemma 1. [2] *Let $\mathbb{F}$ be a field of size q and characteristic p, and d be any integer such that $d+2$ is not divisible by p. The tag generation function $f_s : \mathbb{F} \times \mathbb{F}^d \to \mathbb{F} \times \mathbb{F}$, such that*

$$f_s(r; m) = (r\ ,\ r^{d+2} + \sum_{i=1}^{d} m_i r^i)$$

gives a family of systematic $(q^d, q^{d+2}, q, \frac{d+1}{q})$-strong AMD codes with effective tag length $\varpi_s^*(\kappa, \nu) \leq 2\kappa + 2\log(\nu/\kappa + 3) + 2$ when $p = 2$. [1]

Combining Theorem 1 and Lemma 1 gives us a family of $(q^d, q^{d+2}, q, \alpha, \frac{d+1}{q^{1-\alpha}})$-strong LLR-AMD codes whose failure probability becomes arbitrarily small by choosing q sufficiently large. The effective tag length of this family, when $p = 2$, is upper bounded as

$$\varpi_s^*(\kappa, \nu, \alpha) \leq \frac{2}{1-\alpha}\left(\kappa + \log(\nu/k + 3)\right) + 2.$$

Below, we provide an optimal weak AMD code construction, whose security is proven in Appendix B.

Theorem 2 (Appendix B). *Let $\mathbb{F}$ be a field of size q and characteristic p, $d \in \mathbb{N}$, and $t \in \{2, 3\}$ be such that $t \neq p$. The tag generation function $f_w : \mathbb{F}^d \to \mathbb{F}$, such that*

$$f_w(m) = \sum_{i=1}^{d} (m_i)^t$$

gives a family of systematic $(q^d, q^{d+1}, \frac{2}{q})$-weak AMD codes with the effective tag length $\varpi_w^*(\kappa, \nu) \leq \kappa + 1$ when $p = 2$.

Applying Theorem 1 to this construction results in a family of $(q^d, q^{d+1}, \alpha, \frac{2}{q^{1-\alpha d}})$-weak LLR-AMD codes. The effective tag length of this code family is generally upper bounded by $\varpi_w^*(\kappa, \nu, \alpha) \leq \frac{\kappa + \alpha\nu + 1}{1-\alpha}$, but becomes as low as $\frac{\kappa}{1-\alpha} + \alpha\nu + 3$ when $1/\alpha$ tends from below to a natural number.

Compare the effective tag lengths of the two LLR-AMD constructions. For the strong code, the tag length remains always logarithmic to ν (hence the message length) regardless of leakage rate α. For the weak code however, the tag length increases linearly with ν when $\alpha \neq 0$, and thus it cannot be negligible to the message length for arbitrarily small decoding failure. This can also be seen comparing the decoding failure probabilities $\frac{1}{q^{1-\alpha}}$ and $\frac{1}{q^{1-\alpha d}}$ for the strong and weak LLR-AMD codes: Letting these terms tend to zero, the two constructions achieve the asymptotic rates of 1 and (at most) $1/\alpha$, respectively. It is crucial to know whether the above rates are the highest achievable. We obtain a positive answer to this question by proving non-trivial (almost) tight lower bounds on the effective tag lengths of weak and strong LLR-AMD code families.

[1] We slightly modified the original code description [2] for consistency reasons. We used r and ν in place of x and u, respectively, and let randomness r be part of the $f_s(.,.)$ function's output.

Theorem 3 (Appendix C). *Any weak, resp. strong, LLR-AMD code family $\mathcal{F}$ has an effective tag length lower bounded as*

$$\varpi^*_{\mathcal{F}}(\kappa, \nu, \alpha) \geq \max\{\tfrac{\kappa}{1-\alpha} - 2\ , \kappa + \alpha\nu - 2\}, \quad \textit{resp.} \quad \varpi^*_{\mathcal{F}}(\kappa, \nu, \alpha) \geq \tfrac{2\kappa}{1-\alpha} - 2. \; (3)$$

The effective tag lengths of the AMD constructions (Theorem 2 and Lemma 1) closely match the lower-bound expressions. This indicates the optimality of those constructions under leakage. Again observe that unlike strong ones, weak LLR-AMD codes cannot achieve more than $1/(1+\alpha)$ asymptotic rate under linear leakage rate of α. We ask whether deterministic LR-AMD coding with higher rate (less redundancy) is possible for other leakage scenarios. This is addressed for the block leakage model in the following section.

4.2 BLR-AMD code construction

Theorem 4 introduces a novel deterministic BLR-AMD construction that is optimal as it achieves the asymptotic rate of 1. The construction can be also used as weak and strong LLR-AMD codes. The reason the code stays secure under block leakage is that its tag generation function is nonlinear to all message blocks, and leftover min-entropy even in one message block suffices to protect against algebraic manipulation. This is in contrast with strong LLR-AMD codes (e.g., Lemma 1) which relies only on the min-entropy of the encoding randomness.

Theorem 4 (Appendix D). *For positive integers q and (odd) d, $\mathbb{F}_{q+1}$ be a field of size $q+1$ with primitive element τ, and G be a $d \times d$ non-singular matrix over $\mathbb{Z}_q$ such that*
- each column of G consists of distinct entries, i.e., $\forall j, i, i' \neq i : \; g_{i,j} \neq g_{i',j}$;
- entries of G (as integers) are at most ψd for constant ψ, i.e., $\forall i, j : \; g_{i,j} \leq \psi d$.
The tag generation function $f_{blr} : \mathbb{Z}_q^d \to \mathbb{F}_{q+1}$, such that

$$f_{blr}(m) = \sum_{i=1}^{d} \tau^{\sum_{j=1}^{d} g_{i,j} m_j \mod q} \in \mathbb{F}_{q+1},$$

gives a systematic $(q^d, (q+1)q^d, \alpha, \frac{\psi d}{q^{1-\alpha}})$-BLR-AMD code.

Remark 2. There are possible ways to construct the matrix G in Theorem 4, e.g., using non-singular circulant matrices [3]. In Appendix H, we give one example of constructing G with $\psi = 3$ when q is prime.

The effective tag length of the above construction for $\mathbb{F}_{q+1}$ of characteristic 2 is

$$\varpi^*_{blr}(\kappa, \nu, \alpha) \leq \frac{\kappa + \log(\psi\nu/\kappa + 3)}{1-\alpha} + 3.$$

4.3 Comparing the three constructions

Figure 2 graphs the effective tag lengths of the three LR-AMD constructions defined by $f_s(.;.)$, $f_w(.)$, and $f_{blr}(.)$ with respect to message length parameter $2^7 \leq \nu \leq 2^{20}$, letting leakage rate $\alpha = 0.49 < 0.5$ and security parameter $\kappa = 128$. For the strong LLR-AMD and the weak BLR-AMD constructions, the tag length stays almost constant (around 520 and 260 bits, respectively). This promises the asymptotic rate of 1 when ν tends to infinity. Of course $f_s(.;.)$ bears around two times redundancy of $f_{blr(.)}$ since it carries the encoding randomness. The minimum possible tag length of the weak LLR-AMD construction, however, grows linearly with ν, leading to an asymptotic rate of 0.66.

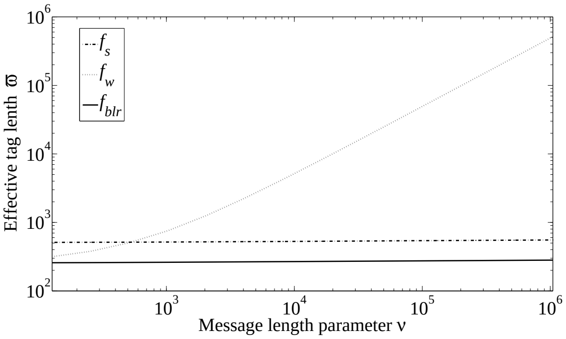


Fig. 2. Comparing the redundancies in the LR-AMD constructions ($\alpha = 0.49$).

5 Wiretap Channels: Manipulation Detection

Consider a special case of Figure 1 when leakage is through a probabilistic wiretapping channel. For a passive wiretapper, Wyner [12] proved that keyless private communication is possible with a slight noise over the wiretapping channel. Keyless manipulation detection however is trivially impossible if the adversary's manipulation power is not restricted. We first study "algebraic" manipulation detection over wiretap channel and next show how coding and modulation can be combined to detect "unrestricted" manipulation over this channel.

5.1 Algebraic manipulation

We consider symmetric and erasure u-ary wiretap channels, defined as follows.

Definition 10 (SWC/EWC). *A (u,p)-symmetric wiretap channel (SWC) transmits codeword as a sequence of elements of set $\mathcal{F}_u$ of size u, such that its wiretapping component, $SC_{u,p}$, either transmits a symbol correctly with probability $1-p$ or corrupts it, i.e., converts to it any other symbol with probability $p/(u-1)$.*
A (u,p)-erasure wiretap channel (EWC) is defined similarly, expect the wiretapping component, $EC_{u,p}$, erases (converts to Λ) symbols instead of corrupting.

When $u = 2$, the definitions lead to the common binary wiretap channels, denoted by p-BEWC and p-BSWC. Observe that the wiretap channel is a special case of linear leakage when leakage is probabilistic, so one may use LLR-AMD codes for them. Applying the construction of Lemma 1 gives the following result.

Corollary 1. *The construction of Lemma 1 detects algebraic manipulation of any message over the (u,p)-EWC with $p > 0.5$, with failure probability*

$$\leq \min_{0.5<\beta<p} \left(\frac{d}{q^{2\beta-1}} + q^{-\frac{(p-\beta)^2}{p\ln(u)}} \right).$$

Here d and q are defined in Lemma 1. The proof of of this result is given as part of the proof for Theorem 5 below. Informally, the upper-bound is calculated as follows: Except with probability $\leq q^{-\frac{(p-\beta)^2}{p\ln(u)}}$, the erasure channel erases β fraction of symbols from the randomness R and the tag $T = f_s(R; m)$, where m is the message. This implies the leftover min-entropy of $1 - \alpha \geq (2\beta - 1)\log q$ for R, and decoding failure of $\leq \frac{d}{q^{2\beta-1}}$. Similarly, the following can be obtained for the weak LLR-AMD construction of Theorem 2

Corollary 2. *The construction Theorem 2 detects algebraic manipulation of a uniform message over the (u,p)-EWC with $p > \frac{d}{d+1}$, with failure probability*

$$\leq \min_{\frac{d}{d+1}<\beta<p} \left(\frac{2}{q^{(d+1)\beta-d}} + q^{-\frac{(d+1)(p-\beta)^2}{2p\ln(u)}} \right).$$

Observe that when $p \leq 0.5$, the LLR-AMD code constructions provide no security guarantees regardless of the value of u. This raises the question of the possibility of tempering detection for $p \leq 0.5$. We show a positive answer through modeling the wiretap channel by block leakage, where only one message block needs to have leftover uncertainty. Theorem 5 proves that the BLR-AMD code construction of Theorem 4 detects algebraic manipulation over a wider range of EWCs, i.e., when $p > 0.5$ or $p^{p^{-1}} > u^{-1}$, which covers e.g., $p > 0.25$ for $u = 2^8$.

Theorem 5 (Appendix E). *The BLR-AMD code construction of Theorem 4, with q such that $\log_u(q+1) \in \mathbb{N}$, detects algebraic manipulation of uniform message over the (u,p)-EWC with failure probability of at most*

$$\epsilon_{blr1} = \min_{0.5<\beta<p} \left(\frac{\psi d}{q^{2\beta-1}} + (q+1)^{-\frac{(p-\beta)^2}{p\ln(u)}} \right) \quad \text{for} \quad p > 0.5, \quad \text{and} \tag{4}$$

$$\epsilon_{blr2} = \min_{\zeta<\beta<p} \left(\frac{\psi d}{q^{\beta}} + (q+1)^{-\frac{(p-\beta)^2}{2p\ln(u)}} + e^{\frac{d}{(q+1)^{\zeta}}} \right) \quad \text{for} \quad p^{p^{-1}} > u^{-1}, \tag{5}$$

where $\zeta = -\log_u(p) < p$. [2]

Proposition 1. *Theorem 5 also holds for (u,p')-SWC with $p' = (1-u^{-1})p$ and p given in the theorem.*

[2] ϵ_{blr2} can be made arbitrarily small, e.g., by choosing $d \approx q^{(\beta+\zeta)/2}$ and q sufficiently large.

Proposition 1 holds since for the codeword X, the adversary's view $Z' = SC_{u,p'}(X)$ can be simulated from the erasure channel output $Z = EC_{u,p}(X)$ by letting $Z'_i = r$ for uniformly random $r \in \mathbb{F}_u$ when $Z_i = \Lambda$, or $Z'_i = Z_i$ otherwise.

The construction of "optimal" AMD codes remains open for wiretap channels that violate the condition on p and u in Theorem 5 and Proposition 1. This includes p-BEWC with $p < 0.5$ and p-BSWC with $p < 0.25$.

5.2 Unrestricted manipulation

We show how the code can be used in practice to detect unrestricted manipulation over tamperable erasure/symmetric wiretap channels. To send a message to Bob, Alice (i) encodes it by the BLR-AMD construction, (ii) applies *Manchester coding*, and (iii) transmits the resulting codeword bit by bit separately via *on-off keying*. The construction does not require any sort of extra randomness (except message/key) in the system. Manchester code is a simple binary error-detecting code that appends to each bit its complement. On-off keying is a popular modulation technique used in digital data communication (esp. fiber optics) which modulates the bit "1" by a carrier wave signal and the bit "0" by the absence of signal. Although the adversary is unrestricted in manipulation, the bitwise nature of the communication leaves her no choice other than tampering with each individual bit using one of the four bitwise functions, i.e., keep, flip, set-to-0, and set-to-1. Proposition 2 formalizes this result.

Proposition 2 (Appendix F.). *Let Enc_{mn}/Dec_{mn} be the Manchester encoding/decoding functions, and f_{blr} be the BLR-AMD code of Theorem 4, where $q = 2^v - 1$ and $\mathbb{F}_{q+1} = GF(2^v)$. The code $Enc_b(m) = Enc_{mn}(m, f_{blr}(m))$* [3] *and*

$$Dec_b(c) = \begin{cases} \hat{m}, & \text{if } Dec_{mn}(c) = (\hat{m}, \hat{t}) \neq \perp, \text{and } \hat{t} = f_{blr}(\hat{m}) \\ \perp, & \text{else} \end{cases}, \tag{6}$$

has code rate almost $\frac{d}{2(d+1)}$ and detects manipulation of uniform message over a p-BEWC (or p/2-BSWC) with $p > 0.5$ with failure probability at most ϵ_{blr1} (as in Theorem 5), if the codeword is sent via on-off keying.

Remark 3. Proposition 2 adopts the plausible assumption that the use of on-off keying prevents the adversary from applying the set-to-0 function, this is because it is theoretically impossible to find a signal that converts both "1" and "0" bit signals to "0". For more details, refer to Appendix I.

5.3 Wiretap codes for active adversaries

We compose the construction of Proposition 2 with wiretap codes [12] for both privacy and integrity of message/key transmission over wiretap channels.

[3] For binary transmission, assume each message block $m_i \in \mathbb{Z}_q$ is mapped to its v-bit string representation before being given to Manchester code (there would be no mapping to 1^v string).

Definition 11. *The code with functions $Enc_w : \{0,1\}^t \to \{0,1\}^k$ and $Dec_w : \{0,1\}^k \to \{0,1\}^t$ is a (t,k,ϵ)-wiretap code over the p-BEWC (resp. p-BSWC) if $\forall m \in \{0,1\}^t : Dec_w(Enc_w(m)) = m$ and for uniform $M \in \{0,1\}^t$ it holds $I(M;Z)/t \leq \epsilon$, where $Z = BEC_p(Enc_w(M))$ (resp. $Z = BSC_p(Enc_w(M))$).*

Proposition 3 (Appendix G). *Let Enc_w/Dec_w denote a (t,k,ϵ)-wiretap code over the p-BEWC (resp. p/2-BSWC), for $p > 0.5$, such that $Enc_w(M)$ is uniform for uniform M. Let Enc_b/Dec_b be the code construction of Proposition 2 with $v \leq t\epsilon$. The code $Enc_{wb}(m) = Enc_b(Enc_w(m))$ and*

$$Dec_{wb}(c) = \begin{cases} Dec_w(Dec_b(c)), & Dec_b(c) \neq \perp \\ \perp, & else \end{cases}. \tag{7}$$

is a $(t,n,2\epsilon)$ wiretap code, with $n = \frac{2k(d+1)}{d}$, which detects manipulation of M over the p-BEWC (or p/2-BSWC) with failure probability at most ϵ_{blr1} (as in Theorem 5), if the codeword is sent via on-off keying.

Known results give (t,k,ϵ)-wiretap code constructions over p-BEWC (resp. p-BSWC) with arbitrarily small $\epsilon > 0$ and of rate arbitrarily close to $1-p$ (resp. $h(p) = -p\log(p) - (1-p)\log(1-p)$) [12]. The above code construction achieves rates arbitrarily close to $(1-p)/2$ (resp. $h(p)/2$) and provides both privacy and integrity of transmissionwith arbitrarily small failure probability.

6 Conclusion

The AMD study in linear and block leakage models captures interesting scenarios of reliable communication in the presence of an adversary who receives arbitrary but bounded leakage about the communication. We proved optimal LLR-AMD and BLR-AMD constructions and showed an application of these codes to manipulation detection over wiretap channels. This work raises a number of directions to future work. These include manipulation detection over more general wiretap channels and finding applications of LR-AMD codes to other areas of cryptography. An example of the latter is adding robustness to non-perfect secret sharing schemes, which is a subject of our ongoing work.

References

1. S. Capkun, M. Cagalj, R. K. Rengaswamy, I. Tsigkogiannis, J. P. Hubaux, and M. Srivastava. Integrity codes: Message integrity protection and authentication over insecure channels. *IEEE Transactions on Dependable and Secure Computing*, 5(4):208–223, 2008.
2. R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. *Advances in Cryptology–EUROCRYPT 2008*, pages 471–488, 2008.
3. P. J. Davis. *Circulant matrices.* Chelsea Publishing Company, 1994.

4. D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.
5. S. Dziembowski and K. Pietrzak. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 293–302, 2008.
6. S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. In *ICS*, pages 434–452, 2010.
7. R. Gennaro, A. Lysyanskaya, T. Malkin, S. Micali, and T. Rabin. Algorithmic tamper-proof (atp) security: Theoretical foundations for security against hardware tampering. In *Theory of Cryptography*, pages 258–277. Springer, 2004.
8. V. Guruswami and A. Smith. Codes for computationally simple channels: Explicit constructions with optimal rate. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 723–732, 2010.
9. R. W. Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.
10. M. Langberg. Oblivious communication channels and their capacity. *IEEE Transactions on Information Theory*, 54(1):424–429, 2008.
11. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.
12. A. D. Wyner. The wire-tap channel. *Bell System Technical Journal*, 54:pp. 1355–1367, 1975.

A Proof of Theorem 1: LLR-AMD

We prove the theorem for strong AMD codes (similar proof can be given for weak AMD codes). Let Enc/Dec denote a $(\mathbf{M}, \mathbf{X}, \mathbf{R}, \epsilon)$-strong AMD code. The security property implies (when there is no leakage)

$$\forall m : \quad \max_{\delta} \Pr_{R}(Dec(Enc(R;m)+\delta) \notin \{m, \perp\}) \leq \epsilon, \tag{8}$$

where R is the uniform randomness of the encoder. For any m and δ, define $\mathcal{R}_{fail}(m,\delta) \subseteq \mathcal{R}$ as the set of r values that lead to the verification failure, by satisfying $Dec(Enc(R;m)+\delta) \notin \{m, \perp\}$. Since R is uniform, the probability that $R \in \mathcal{R}_{fail}(m,\delta)$ equals to $|\mathcal{R}_{fail}(m,\delta)|/\mathbf{R}$; thus, to write (8) as $\forall m : \quad \max_\delta |\mathcal{R}_{fail}(m,\delta)| \leq \epsilon \mathbf{R}$. Let Z be any random variable such that the randomness R is $(1-\alpha)$-weak conditioned on Z for $0 \leq \alpha \leq 1$, i.e., $E_z\left(\max_r \Pr(R=r|Z=z) \leq \mathbf{R}^{\alpha-1}\right)$. For any message m, the probability of failure when Z is leaked to the adversary $\mathcal{A}dv$ is upper bounded as

$$\begin{aligned}
&\Pr(Dec(Enc(R;m)+\mathcal{A}dv(Z)) \notin \{m,\perp\}) = E_z\left(\Pr(Dec(Enc(R;m)+\mathcal{A}dv(z)) \notin \{m,\perp\}|Z=z)\right) \\
&\quad \leq E_z\left(\max_\delta \Pr(R \in \mathcal{R}_{fail}(m,\delta) \mid Z=z)\right) \leq E_z\left(\max_\delta |\mathcal{R}_{fail}(m,\delta)| \max_r \Pr(R=r|Z=z)\right) \\
&\quad = \max_\delta |\mathcal{R}_{fail}(m,\delta)| \quad E_z\left(\max_r \Pr(R=r|Z=z)\right) \leq \epsilon \mathbf{R}^{\alpha}.
\end{aligned}$$

B Proof of Theorem 2: weak AMD

We shall show that for the uniform message $M \in \mathbb{F}^d$ and any $(\delta_m, \delta_t) \in \mathbb{F}^d \times \mathbb{F}$ such that $\delta_m \neq 0$, it holds $\Pr_M(f_w(M+\delta_m) = f_w(M) + \delta_t) \leq \frac{2}{q}$. Since $\delta_m = (\delta_{m,1}, \ldots, \delta_{m,d}) \neq 0$, there exists at least non-zero one element $\delta_{m,o} \neq 0$ for $1 \leq o \leq d$. This lets us write the term $f_w(M+\delta_m) - f_w(M) - \delta_t$ as a polynomial of degree $t-1$ with respect to the variable M_o, i.e., $Poly(M_o) \overset{\Delta}{=}$

$$f_w(M+\delta_m) - f_w(M) - \delta_t = \left[\sum_{i=1}^{d} (M_i + \delta_{m,i})^t - M_i^t\right] - \delta_t = \sum_{j=1}^{t} \binom{t}{j} \delta_{m,o}^j M_o^{t-j} + a_0,$$

where $a_0 = \left[\sum_{i=1, i\neq o}^{d} (M_i + \delta_{m,i})^t - M_i^t\right] - \delta_t$ is the constant term. For any values of $(M_i)_{i\neq o}$, hence fixed a_0, the polynomial $Poly(M_o)$ evaluates to zero for at most $t-1 \leq 2$ (out of q) values of M_o. The polynomial thus becomes zero with probability at most $(t-1)/q \leq 2/q$, implying the failure probability bound.

The effective tag length of this code family when $p = 2$ is obtained as follows. For integers $\kappa, \nu \in \mathbb{N}$, let $q = 2^{\kappa+1}$ and $d = \lceil \nu / \log q \rceil$ so that both $\epsilon = 2/q \leq 2^{-\kappa}$ and $|\mathcal{F}^d| = q^d \geq 2^\nu$ are satisfied. By restricting the source space $\mathcal{F}^d$ to only $\mathbf{M} = 2^\nu$ elements the code range will also reduce to $\mathbf{X} = q2^\nu$ elements in $\mathcal{F}^{d+1}$. This leads to $\log \mathbf{X} - \nu = \nu + \log q - \nu = \kappa + 1$.

C Proof of Theorem 3: tag length

The proof relies on the results of the following lemma.

Lemma 2. *For any weak, resp. strong, LLR-AMD code the failure probability is lower bounded as*

$$\epsilon \geq \max\{\left((1-e^{-1})\frac{\mathbf{M}-1}{\mathbf{X}-1}\right)^{1-\alpha} \ , \ (1-e^{-1})\mathbf{M}^\alpha \frac{\mathbf{M}-1}{\mathbf{X}-1}\}, \tag{9}$$

$$\textit{resp.} \quad \epsilon \geq \left((1-e^{-1})\frac{\mathbf{M}-1}{\mathbf{X}-1}\right)^{(1-\alpha)/2}. \tag{10}$$

Proof. We start by the $(\mathbf{M}, \mathbf{X}, \alpha, \epsilon)$-weak LLR-AMD code. We shall show that for any such code there exists a message distribution $M \in \mathcal{M}$, a leakage variable Z with $\tilde{H}_\infty(M|Z) \geq (1-\alpha)\log \mathbf{M}$, and an adversary whose success chance in changing M is lower bounded by (9). We choose M to be uniform and Z to be an $\alpha \log \mathbf{M}$-bit string that represents answers to the adversary's $\alpha \log \mathbf{M}$ questions about the codeword. The variable Z is such that each bit Z_i is defined by $Z_i = Query_i(Z_1^{i-1}, M)$, where $Query_i$ shows the i-th question. Let $X = Enc(M)$ be the codeword for M. The adversary can choose any non-zero adversarial noise $\delta \in \mathcal{X}/\{0\}$ to be added to the X. There are $n = \mathbf{X} - 1$ values for δ, at least $t = M - 1$ of which lead to valid codewords $X + \delta$. Let $\mathcal{X}^+$ be the set of such valid δ values. If the adversary picks δ randomly, her success chance will be $\geq t/n$. We now describe the adversary's strategy as follows. She first chooses a random subset $\mathcal{H}_0 \subseteq \mathcal{X}/\{0\}$ of size $k = n/t$ and runs the following algorithm.

```
$\mathcal{H} \leftarrow \mathcal{H}_0$.
for $(i = 1$ to $\alpha \log \mathbf{M})$
    Partition $\mathcal{H}$ arbitrarily to $\mathcal{H}_1$ and $\mathcal{H}_2$ of (almost) equal sizes.
    Set $Z_i \leftarrow$ whether $|\mathcal{H}_1 \cap \mathcal{X}^+| > 0$.
if $Z_i = 1$ (Yes) then $\mathcal{H} \leftarrow \mathcal{H}_1$.
else $\mathcal{H} \leftarrow \mathcal{H}_2$.
return $\delta$ that is randomly chosen from $\mathcal{H}$.
```

The size of $\mathcal{H}$ at the end of the algorithm decreases to $k/\mathbf{M}^\alpha$. The adversary succeeds with probability $\mathbf{M}^\alpha/k$ if and only if $\mathcal{H}_0 \cap \mathcal{X}^+$ is not empty, whose probability is obtained as

$$\Pr(|\mathcal{H}_0 \cap \mathcal{X}^+| > 0) = 1 - \Pr(|\mathcal{H}_0 \cap \mathcal{X}^+| = 0) = 1 - \frac{\binom{n-t}{k}}{\binom{n}{k}}$$
$$= 1 - \frac{(n-k) \times \cdots \times (n-k-t)}{n \times \cdots \times (n-t)} \geq 1 - (1 - k/n)^t = 1 - (1 - 1/t)^t \geq 1 - e^{-1}.$$

This concludes the adversary's success probability is at least $\epsilon \geq (1-e^{-1})\mathbf{M}^\alpha/k = (1-e^{-1})\mathbf{M}^\alpha \frac{\mathbf{M}-1}{\mathbf{X}-1}$, which is the second term of (9). For the first term, we use the fact that the message size $\mathbf{M}$ is such that after $\alpha \log \mathbf{M}$ questions the adversary cannot guess the correct message with probability more than ϵ, and this implies $\mathbf{M}^{1-\alpha} \geq 1/\epsilon$. We use this to write (noting that $0 \leq \alpha \leq 1$)

$$\epsilon^{1/(1-\alpha)} \geq (1-e^{-1})\frac{\mathbf{M}-1}{\mathbf{MT}-1} \implies \epsilon \geq \left((1-e^{-1})\frac{\mathbf{M}-1}{\mathbf{X}-1}\right)^{1-\alpha}.$$

A similar argument can be used for the $(\mathbf{M}, \mathbf{X}, \mathbf{R}, \alpha, \epsilon)$-strong LLR-AMD code: For uniform randomness R and the variable Z such that $\tilde{H}_\infty(R|Z) \geq (1-\alpha)\log \mathbf{R}$, the adversary can use a similar strategy to Algorithm 1 with $\alpha \log \mathbf{R}$ questions to achieve the success chance of $\epsilon \geq (1-e^{-1})\mathbf{R}^\alpha \frac{\mathbf{R}(\mathbf{M}-1)}{\mathbf{X}-1}$, noting that there are at least $\mathbf{R}(\mathbf{M}-1)$ valid δ values in $\mathcal{H}_0$. In a strong LLR-AMD code, the adversary is assumed to know the message. So the randomness size $\mathbf{R}$ should be large enough to satisfy $\mathbf{R}^{1-\alpha} \geq 1/\epsilon$. Combining this with the above shows the following for $0 \leq \alpha \leq 1$ which proves (10).

$$\epsilon^{2/(1-\alpha)} \geq (1-e^{-1})\frac{\mathbf{M}-1}{\mathbf{X}-1} \implies \epsilon \geq \left((1-e^{-1})\frac{\mathbf{M}-1}{\mathbf{X}-1}\right)^{(1-\alpha)/2}. \Box$$

We use (9) to bound the effective tag length of weak AMD code families as

$$\log \mathbf{X} - \nu \geq \log \frac{\mathbf{X}}{\mathbf{M}} = \log \left(\frac{\mathbf{X}}{\mathbf{M}-1} \times \frac{\mathbf{M}-1}{\mathbf{M}}\right) \geq \log \frac{\mathbf{X}-1}{\mathbf{M}-1} + \log(1 - \mathbf{M}^{-1})$$
$$\geq \max\{\frac{1}{1-\alpha}\log\frac{1}{\epsilon}\ ,\ \log\frac{1}{\epsilon} + \alpha \log \mathbf{M}\} + \log(1-e^{-1}) + \log(1-\mathbf{M}^{-1})$$
$$\geq \max\{\frac{\kappa}{1-\alpha}\ ,\ \kappa + \alpha\nu\} - 2.$$

Similarly, (10) is used to bound the effective tag length of strong code families

$$\log \mathbf{X} - \nu \geq \frac{2}{1-\alpha}\log\frac{1}{\epsilon} + \log(1-e^{-1}) + \log(1-\mathbf{M}^{-1}) \geq \frac{2\kappa}{1-\alpha} - 2.$$

D Proof of Theorem 4: BLR-AMD

The code construction Enc_{blr}/Dec_{blr} is systematic, so we only need to show the security property. Let the message $M \in \mathbb{Z}_q^d$ and Z follow the block leakage model such that for some $o \in \{1, \dots, d\}$ it holds that $\tilde{H}_\infty(M_o|Z, (M_j)_{j \neq o}) \geq (1-\alpha)\log q$. The decoding failure probability when Z is leaked to the adversary $\mathcal{A}dv$ is upper bounded as

$$\begin{aligned}
&\Pr_M(Dec_{blr}(Enc_{blr}(M) + \mathcal{A}dv(Z)) \notin \{M, \perp\}) \\
&= E_z\left(\Pr_M(Dec_{blr}(Enc_{blr}(M) + \mathcal{A}dv(z)) \notin \{M, \perp\}|Z = z)\right) \\
&\leq E_z\left(\max_\delta \Pr_M(Dec_{blr}(Enc_{blr}(M) + \delta) \notin \{M, \perp\}|Z = z)\right) \\
&\overset{(b)}{=} E_z\left(\max_{\delta_m \neq 0, \delta_t} E_{(m_j)_{j\neq o}|Z=z}\big(\Pr_{M_o}(f_{blr}(M + \delta_m) = f_{blr}(M) + \delta_t|Z = z, (M_j = m_j)_{j \neq o})\big)\right)
\end{aligned} \tag{11}$$

Equality (a) follows from the law of total probability and the systematic construction of the BLR-AMD code. For fixed $(M_j = m_j)_{j\neq o} \in \mathbb{Z}_q^{d-1}$, $\delta_m \in \mathbb{Z}_q^d$, and $\delta_t \in \mathbb{F}_{q+1}$, we write the term $f_{blr}(M + \delta_m) - f_{blr}(M) - \delta_t$ as

$$\begin{aligned}
\sum_{i=1}^d \left[\tau^{\sum_j g_{i,j}(M_j+\delta_{m,j})} - \tau^{\sum_j g_{i,j}M_j}\right] - \delta_t = \sum_{i=1}^d \left[\left(\tau^{\sum_j g_{i,j}\delta_{m,j}} - 1\right)\tau^{\sum_{j\neq o} g_{i,j}m_j}\tau^{g_{i,o}M_o}\right] \\
-\delta_t = \sum_{i=1}^d \left[a_i Y^{g_{i,o}}\right] + a_0 \triangleq P_{\delta,(m_j)_{j\neq o}}(Y),
\end{aligned} \tag{12}$$

letting $a_0 = -\delta_t$, $Y = \tau^{M_o}$, and a_i be the coefficient of $Y^{g_{i,o}}$ in the summation, i.e., $a_i = \left(\tau^{\sum_j g_{i,j}\delta_{m,j}} - 1\right)\tau^{\sum_{j\neq o} g_{i,j}m_j}$. Applying this to (11), we need to find an upper-bound on

$$E_z\left(\max_{\delta_m \neq 0, \delta_t} E_{(m_j)_{j\neq o}|Z=z}\big(\Pr_{M_o}(P_{\delta,(m_j)_{j\neq o}}(Y) = 0|Z = z, (M_j = m_j)_{j\neq o})\big)\right). \tag{13}$$

The polynomial $P_{\delta,(m_j)_{j\neq o}}(Y)$ is of degree at most $\max_i(g_{i,o}) \leq \psi d$ over $\mathbb{F}_{q+1}$. Lemma 3 shows that the polynomial is non-constant since it has at least one non-zero coefficient.

Lemma 3. *For any choice of message blocks $(M_j = m_j)_{j\neq o}$, $\delta_m \neq 0$, and δ_t, the polynomial $P_{\delta,(m_j)_{j\neq o}}(Y)$ has at least one non-zero coefficient.*

Proof. We prove the claim by contradiction. Assume that all a_i's are zero, implying (τ is a primitive element in $\mathbb{F}_{q+1}$)

$$\forall 1 \leq i \leq d: \quad \left(\tau^{\sum_j g_{i,j}\delta_{m,j}} - 1\right)\tau^{\sum_{j\neq o} g_{i,j}m_j} = 0 \in \mathbb{F}_{q+1} \Rightarrow \sum_{j=1}^d g_{i,j}\delta_{m,j} = 0 \in \mathbb{Z}_q.$$

The above can be written as $\delta_m.G = 0$ over $\mathbb{Z}_q$, which holds only if $\delta_m = 0$ as G is non-singular. This contradicts the adversarial assumption $\delta_m \neq 0$. □

For any δ (such that $\delta_m \neq 0$) and $(M_j = m_j)_{j \neq o}$, at most ψd values of Y (hence M_o) make the polynomial evaluate to zero. Let $\mathcal{M}_{o,fail}(\delta, (m_j)_{j\neq o})$ of size at most ψd be the set of such M_o values that lead to decoding failure, implying

$$P_{\delta,(m_j)_{j\neq o}}(Y) = 0 \iff M_o \in \mathcal{M}_{o,fail}(\delta, (m_j)_{j\neq o}).$$

We prove security by upper-bounding the failure probability (13) as follows.

$$\begin{aligned}
&E_z\left(\max_{\delta_m\neq 0,\delta_t} E_{(m_j)_{j\neq o}|Z=z}\big(\Pr_{M_o}(P_{\delta,(m_j)_{j\neq o}}(Y)=0|Z=z,(M_j=m_j)_{j\neq o})\big)\right)\\
&= E_z\left(\max_{\delta_m\neq 0,\delta_t} E_{(m_j)_{j\neq o}|Z=z}\big(\Pr_{M_o}(M_o \in \mathcal{M}_{o,fail}(\delta,(m_j)_{j\neq o})|Z=z,(M_j=m_j)_{j\neq o})\big)\right)\\
&\leq E_z\left(\max_{\delta_m\neq 0,\delta_t} E_{(m_j)_{j\neq o}|Z=z}\big(|\mathcal{M}_{o,fail}(\delta,(m_j)_{j\neq o})| \max_{m_o}\Pr_{M_o}(M_o=m_o|Z=z,(M_j=m_j)_{j\neq o})\big)\right)\\
&\overset{(a)}{\leq} \psi d E_z\left(\max_{\delta_m\neq 0,\delta_t} E_{(m_j)_{j\neq o}|Z=z}\big(\max_{m_o}\Pr_{M_o}(M_o=m_o|Z=z,(M_j=m_j)_{j\neq o})\big)\right)\\
&\overset{(b)}{=} \psi d E_z\left(E_{(m_j)_{j\neq o}|Z=z}\big(\max_{m_o}\Pr_{M_o}(M_o=m_o|Z=z,(M_j=m_j)_{j\neq o})\big)\right)\\
&\overset{(c)}{=} \psi d E_{z,(m_j)_{j\neq o}}\big(\max_{m_o}\Pr_{M_o}(M_o=m_o|Z=z,(M_j=m_j)_{j\neq o})\big)\\
&\overset{(d)}{\leq} \frac{\psi d}{q^{1-\alpha}}.
\end{aligned}$$

Inequality (a) holds since we have $|\mathcal{M}_{o,fail}(\delta,(m_j)_{j\neq o})| \leq \psi d$, equality (b) is attained by removing $\max_\delta$ as the expression has become independent of this parameter, equality (c) uses the law of total probability, and inequality (d) follows the assumption that $\tilde{H}_\infty(M_o|Z,(M_j)_{j\neq o}) \geq (1-\alpha)\log q$.

E Proof of Theorem 5

For uniform message $M \in \mathbb{Z}_q^d$, let $T = f_{blr}(M) \in \mathbb{F}_{q+1}$ denote the tag calculated by the BLR-AMD code and $X = (M,T) = (X_1, \ldots, X_{d+1})$ denote the codeword. Let $\eta = \log_u(q+1) \in \mathbb{N}$. For the purpose of u-ary transmission over (u,p)-EWC, we replace each message block in the codeword by a sequence of η symbols over $\mathcal{F}_u$; hence, each codeword element X_i consists of η channel symbols. The theorem provides two bounds, namely ϵ_{blr1} (4) and ϵ_{blr2} (5), on the BLR-AMD detection failure probability under two different conditions of $p > 0.5$ and $p^{p^{-1}} > u^{-1}$, respectively. To prove the two bounds, we provide different approaches to bounding the failure probability of the code.

Approach 1: Proving ϵ_{blr1} in (4) for $p > 0.5$. Considering $0.5 < \beta < p$, any message block M_o for $o \in \{1, \ldots, d\}$, and the tag T, we shall study two events: $\mathcal{E}_1$ that the channel leakage leaves $(2\beta - 1)\log(q)$ bits of leftover min-entropy in M_o and $\mathcal{E}_2$ that the BLR-AMD decoder detects adversarial tampering (assuming $\mathcal{E}_1$ holds). The failure probability will be then bounded as $\epsilon_{blr1} \leq \Pr(\overline{\mathcal{E}_1}) + \Pr(\overline{\mathcal{E}_2})$.

Let η_o and η_t be the numbers of symbols erased from M_o and T, respectively. We have from the chain rule of min-entropy

$$\tilde{H}_\infty(M_o|Z,(M_i)_{i\neq o}) \geq \tilde{H}_\infty(M_o|(M_i)_{i\neq o}) - (\eta - \eta_t)\log(u) = (\frac{\eta_o + \eta_t}{\eta} - 1)\log(q).$$

Noting that $\Pr(\overline{\mathcal{E}_1}) = \Pr(\eta_o + \eta_t < 2\beta\eta)$, we obtain this probability as

$$\Pr(\overline{\mathcal{E}_1}) = \sum_{i=0}^{\lfloor 2\beta\eta \rfloor} \binom{2\eta}{i} p^i (1-p)^{2\eta - i} \leq e^{-\frac{(p-\beta)^2}{2p} 2\eta} = e^{-\frac{(p-\beta)^2}{p} \log_u(q+1)} = (q+1)^{-\frac{(p-\beta)^2}{p \ln(u)}},$$

where the inequality follows the Chernoff bound. When $\mathcal{E}_1$ holds, the leftover min-entropy of M_o shows the uncertainty rate of $1-\alpha \geq 2\beta - 1$. From Theorem 4, the BLR-AMD decoder fails with probability $\Pr(\overline{\overline{\mathcal{E}_2}}) \leq \frac{\psi d}{q^{2\beta-1}}$. Proof is completed.

Approach 2: Proving ϵ_{blr2} in (5) for $p^{p^{-1}} > u^{-1}$. The condition on p implies $p > \zeta$ for $\zeta = \log_u(1/p)$. Choosing $\zeta < \beta < p$, we consider three events: $\mathcal{E}_1$ that there is (at least) one message block M_o, $o \in \{1, \ldots, d\}$ that is completely erased, $\mathcal{E}_2$ that at least $\beta\eta$ symbols are erased from the tag T, and $\mathcal{E}_3$ that the BLR-AMD decoder detects adversarial tampering (assuming that $\mathcal{E}_1$ and $\mathcal{E}_2$ hold). The overall failure probability is bounded as $\epsilon_{blr2} \leq \Pr(\overline{\mathcal{E}_1}) + \Pr(\overline{\mathcal{E}_2}) + \Pr(\overline{\mathcal{E}_3})$.

A message block M_i is completely erased with probability $p' \geq p^\eta = p^{\log_u(q+1)} = (q+1)^{\log_u(p)} = (q+1)^{-\zeta}$. This implies $\Pr(\overline{\mathcal{E}_1}) = (1-p')^d \leq e^{-p'd} = e^{-\frac{d}{(q+1)^\zeta}}$. On the other hand, $\mathcal{E}_2$ holds except with probability

$$\Pr(\overline{\mathcal{E}_2}) = \sum_{i=0}^{\lfloor \beta\eta \rfloor} \binom{\eta}{i} p^i (1-p)^{2\eta - i} \leq e^{-\frac{(p-\beta)^2}{2p} \eta} = (q+1)^{-\frac{(p-\beta)^2}{2p \ln(u)}}.$$

Provided that $\mathcal{E}_1$ and $\mathcal{E}_2$ holdd, the leftover min-entropy of M_o is bounded as

$$\tilde{H}_\infty(M_o | Z, (M_i)_{i \neq o}) \geq \tilde{H}_\infty(M_o | (M_i)_{i \neq o}) - (1-\beta)\eta \log(u) = \beta \log(q),$$

which implies the uncertainty rate of $1-\alpha \geq \beta$ and BLR-AMD decoding failure probability of $\Pr(\overline{\mathcal{E}_3}) \leq \frac{\psi d}{q^\beta}$ (from Theorem 4). This completes the proof.

F Proof of Proposition 2

The code rate is the product of the rates of the Manchester code, 0.5, and the BLR-AMD code, which is almost $\frac{d}{d+1}$ (there is also a factor of $\frac{\log(q)}{\log(q+1)}$ that is close to 1). We moreover show that the failure probability of the code Enc_b/Dec_b is precisely that of the BLR-AMD code over p-BEWC (or $p/2$-BSWC), which equals ϵ_{blr1} for $p > 0.5$. We show this by discussing that using on-off keying and Manchester coding causes a bitwise manipulation adversary to be either detected or behave like an additive (keep and flip) adversary, whose manipulation is detected by the BLR-AMD code from Theorem 5. For message M, we denote the n-bit codeword $X = Enc_b(M)$, where $n = 2(d+1)v$, by $X = (X_1, X_2, \ldots, X_n)$. The on-off keying transmission makes the adversary only choose from keep, flip, and set-to-1 functions. Assume such an adversary wants to tamper with the codeword and let $Tamp_A = (t_1, t_2, \ldots, t_n)$ be the sequence of bit-manipulation functions over the set of keep, flip, and set-to-1. We claim that $Dec_{mn}(Tamp_A(X)) \in \{\perp, Dec_{mn}(Tamp_S(X))\}$, where $Tamp_S = (t'_1, t'_2, \ldots, t'_n)$ is an "additive" manipulation sequence such that $\forall 1 \leq i \leq n/2 : (t'_{2i-1}, t'_{2i}) =$

$$\begin{cases} (\text{keep, keep}), & (t_{2i-1}, t_{2i}) \in \{(\text{keep, set-to-1}), (\text{set-to-1, keep}), (\text{set-to-1, set-to-1})\} \\ (\text{flip, flip}), & (t_{2i-1}, t_{2i}) \in (\text{flip, set-to-1}), (\text{set-to-1, flip})\} \\ (t_{2i-1}, t_{2i}), & \text{else} \end{cases} \qquad (14)$$

We consider the case where $Dec_{mn}(Tamp_A(X)) \neq \perp$ since otherwise we are done with the proof. For every $1 \leq i \leq n/2$, the pair of codeword bits (X_{2i-1}, X_{2i}) are either 01 or 10. We prove the claim by showing in both of these cases $(t'_{2i-1}(X_{2i-1}), t'_{2i}(X_{2i})) = (t_{2i-1}(X_{2i-1}), t_{2i}(X_{2i}))$. We show the equality for $(X_{2i-1}, X_{2i}) = 01$ and the other case can be argued similarly: The equality holds trivially from (14) if the pair (t_{2i-1}, t_{2i}) does not include any set-to-1 function; if not, the only valid options are $(t_{2i-1}, t_{2i}) \in \{(\text{keep, set-to-1}), (\text{set-to-1, flip})\}$ for which the equality again holds.

G Proof of Proposition 3

For parameters d and v of the BLR-AMD code, let $n = 2(d+1)v$ and $k = dv$. The codeword $C = Enc_{wb}(M)$ is obtained by applying three encoding functions sequentially. The first (wiretap) encoding gives $X = Enc_w(M) \in \{0,1\}^k$ which is uniform for the uniform message $M \in \{0,1\}^t$. The second (BLR-AMD) encoding gives $Y = (X, f_{blr}(X)) \in \{0,1\}^{n/2}$, and the third (Manchester) encoding results in $C = Enc_{mn}(Y)$. The code rate is $t/n = (td)/(2k(d+1))$. The detection failure probability equals that of the code Enc_b/Dec_b and uniformity of X (see Proposition 2). It remains to prove the privacy property of the code.

We prove privacy for p-BEWC (noting that it also works for $p/2$-BSWC). Manchester encoder Enc_{mn} appends to each bit of Y its negation. If both a bit and its negation are erased by p-BEWC (which occurs with probability $p' = p^2$), Eve cannot discover the bit. This implies that Eve's view $Z = BEC_p(C)$ can be built from $Z' = BEC_{p'}(Y)$, i.e., the view over the p'-BEC without Manchester coding. We thus remove Manchester coding and assume that Eve's view is $Z' = (Z'_1, Z'_2)$, where $Z'_1 = BEC_{p'}(X)$ and $Z'_2 = BSC_{p'}(f_{blr}(X))$. We conclude

$$\begin{aligned} I(M;Z) &= I(M;Z'_1,Z'_2) = I(M;Z'_1) + I(M;Z'_2|Z'_1) \leq I(M;Z'_1) + H(Z'_2) \\ &\leq I(M;Z'_1) + (n/2-k) \leq I(M;Z'_1) + v \quad \Rightarrow \quad I(M;Z)/t \leq \epsilon + v/t \leq 2\epsilon. \end{aligned}$$

H Non-singular matrix construction

Let H be a $d \times d$ diagonal matrix over (field) $\mathbb{Z}_q$, where q is prime and $d < 3q$, with entries $H_{i,i} = i$ for $1 \leq i \leq d$. The following algorithm converts H into a non-singular matrix that has non-identical entries in each and every column. It is easy to show that the value of s is always upper bounded by $2i$ and thus at the end, all entries in resulting matrix are less or equal to $2d + d = 3d$.

```
$G \leftarrow H$
**for** $\big(j = 1$ to $d-1\big)$
    Add column $j$ of $G$ to its column $j+1$.
$s \leftarrow 2$
**for** $\big(i = 2$ to $d\big)$
    **while** $\big(s$ equals any entry of $G$ up to row $i-1\big)$
        $s \leftarrow s+1$
    Add $s$ times the first row of $G$ to row $i$.
**return** $G$
```

I On-off keying

On-off keying is the simplest form of amplitude-shift keying (ASK) that transmits the bit “1” as the presence a carrier wave signal and the bit “0” as the absence of the signal. The carrier wave is usually a high frequency sinusoidal signal that is trimmed for a relatively short time interval. To demodulate a received signal, the signal energy is obtained and compered to a threshold value: Below the threshold indicates “0“ and above it indicates “1”. We assume that the carrier wave is fixed and public to all the parties (including Eve). Although on-off keying is in essence a binary modulation, it can work with any underlying modulation scheme by letting “0” be the absence of signal and “1” be transmitted as a publicly known (fixed) modulated signal. Manipulation of a bit (transmitted by on-off keying) is by injecting an adversarial signal to the channel. Assume that the carrier wave is one period of the sine signal. As illustrated in Table 1, there are appropriately-shaped signals to realize the keep, flip, and set-to-1 functions. However, it is not possible to realize a (deterministic) set-to-0 for a bit since there is no signal to annihilate the energy of both “0” and “1” signals. Of course, the adversary could set a transmitted bit to 0 if she knew it by either keeping or flipping the bit (this is not considered as set-to-0). This property lets us replace, without loss of generality, the unlimited bitwise manipulation adversary with an additive-and-set-to-1 adversary.

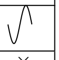
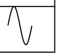
Transmission		Tampering	
bit abstraction	signal	bit abstraction	signal
0	—	keep	—
		flip	
1		set-to-0	×
		set-to-1	

Table 1. Bitwise manipulation for on-off keying.