GLEVIAN and VIGORNIAN: Robust beyond-birthday AEAD modes

Peter Campbell
National Cyber Security Centre
Peter.C@ncsc.gov.uk

Version 1.0; Published 13 September 2023

Abstract

The National Cyber Security Centre (NCSC) is the government organisation responsible for mitigating cyber security risks to the UK. Our work securing UK public- and private-sector networks involves (amongst many other security measures) research into cryptographic design, primarily to protect data requiring long-term security or data for which we have a particularly low tolerance of risk to its transmission and storage. Our algorithms prioritise robustness over other important considerations, such as performance, more highly than other designs.

We present GLEVIAN and VIGORNIAN: two AEAD modes with proofs of beyond-birthday security, security against nonce misuse, and against the release of unverified plaintext – both of the latter in strong notions of these security properties. We discuss our hierarchy of requirements for AEAD modes, and the rationale for the design choices made.

GLEVIAN and VIGORNIAN demonstrate we can achieve significantly improved robustness over GCM for use cases where some performance degradation is acceptable. We are not aware of other designs offering exactly the security properties of GLEVIAN and VIGORNIAN, and are publishing our designs to support the research that will inform the recently announced effort by NIST to standardise new modes of operation. We believe our work could be of interest to those with use cases similar to ours, and we offer suggestions for future research that might build on the work in this paper.

Keywords: authenticated encryption, block cipher, mode of operation, requirements, robustness.

1 Introduction

The mission of the National Cyber Security Centre (NCSC) is to make the UK the safest place to live and work online. As part of our remit we act as the UK Government's technical authority on cryptography. We provide assessment of cryptography for UK Government use cases, in addition to offering consultancy on cryptography across government and guidance in the public sphere.

We also conduct research into cryptographic design, principally for *high-threat* use cases in which data might require very long-term security, or have some sensitivity to it that warrants a particularly risk-averse approach to its handling. This research covers the design and implementation of cryptography used to protect that data. For high-threat use cases we place great value on reducing the likelihood that data is put at risk though misuse of cryptographic algorithms and protocols, since NCSC often does not have direct oversight of the end-users of the cryptography we design. In designing for these use cases we are usually able to make design trade-offs to prioritise robustness and simplicity over other important considerations, such as performance.

In the last few years our symmetric cryptography research has included block cipher modes of operation for Authenticated Encryption with Associated Data (AEAD); henceforth, modes. We designed new modes GLEVIAN and VIGORNIAN[1] which offer security features that were not available in existing modes. They offer nonce-misuse resistance, security against the release of unverified plaintext (RUP) and beyond-birthday security (when not misused) in the standard model.

Practical considerations around providing a near drop-in replacement for GCM also influenced our designs; neither mode is tied to the use of a specific block cipher.

We discuss the properties we target and our rationale for their selection in detail in Section 2. While complete modes in their own right, GLEVIAN and VIGORNIAN also serve as proof of concept that it is possible to provide these desirable features with a practical and concrete design. We drew heavily on prior academic work in our designs, which are a combination of components first described by other cryptographers. GLEVIAN and VIGORNIAN are less efficient than GCM, but sufficiently performant in our use cases, which include both extremely high-bandwidth datalinks and resource-constrained devices.

The National Institute of Standards and Technology (NIST) recently announced their third workshop on block cipher modes of operation, to "discuss how NIST can best address the limitations of the [modes] that are approved in the NIST Special Publication 800-38 Series" [NIS23]. We are publishing GLEVIAN and VIGORNIAN, with security proofs for the properties we target, as a contribution to the body of research that will support NIST's future standardisation efforts. We believe our work could be of interest to those who have use cases where robustness and beyond-birthday security are high priorities. We also suggest some future research directions, which could lead to improved designs that draw on our work.

1.1 Structure

In Section 2, we describe various desirable features a mode might have and offer our own perspective on the relative importance of each design goal. We justify our opinions partly as a primary source and partly by reference to the academic literature. We discuss some existing standardised and well-studied robust unstandardised AEAD schemes, and the extent to which they meet our requirements. In Section 3, we introduce GLEVIAN and VIGORNIAN, explain how they are a combination of existing constructions from the academic literature and make our security claims. We then provide detailed proofs of our security claims for GLEVIAN and VIGORNIAN in Section 4 and its supporting appendices.

1.2 Acknowledgements

The author is grateful for research input from colleagues in the NCSC cryptography teams in writing this document.

[1] The names GLEVIAN and VIGORNIAN have been chosen to resemble the Roman names for the cities of Gloucester (Glevum) and Worcester (Vigornia), both close to Cheltenham where NCSC has a large presence; and to sound similar to the names of musical *modes*.

2 Requirements and Priorities

An AEAD scheme is a symmetric communication scheme in which traffic is confidential and authentic. A block cipher mode of operation (or simply, mode) is a method for using a fixed-width block cipher to process data, and for the purpose of this paper that will mean making an AEAD scheme that is capable of processing variable-width message and AD inputs.

All AEAD designs involve trade-offs between the costs of security properties and efficiency. In this section we will discuss how we balance these costs and features for our use cases. After introducing some general themes, we will discuss more specific properties that a mode may (or may not) satisfy. The general themes we consider are:

Security We take a conservative approach to the setting of security parameters; in particular we carefully consider the security of AEAD deployments that use multiple keys at once, and which process high quantities of data. For this reason, we aim to quantify the multi-user security of our modes, and require beyond-birthday security.

Robustness We design for high-threat use cases and take a risk-averse approach to the design, implementation, and deployment of cryptography. In complex deployments, it is challenging to guarantee cryptography is always used as the designers intended, so we greatly value features that make algorithms resilient to misuse; that is, they are *secure by design*. In particular, we value strong notions of nonce-misuse resistance and security against the Release of Unverified Plaintext (RUP).

Compatibility Wide support for hardware-accelerated AES computations in common computing platforms makes it desirable for our use cases to continue to support AES. However, to allow the widest deployability and to future-proof the mode, we require a mode that works with any block cipher that has an appropriate key and block size.

2.1 High priority requirements

2.1.1 Multi-user security

We are designing for systems made up of many users who are using a mode under different keys. We require the system to be secure against an adversary willing to mount an attack that compromises any one of those many users, even if the adversary has limited control over which user it compromises. We also require security of the system to hold when these users are communicating high quantities of data, for example on high-bandwidth links. Furthermore, algorithms can be in use for a long time, and we assume data-rates tend to increase over time, so we make conservative assessments on how much data is theoretically available to an adversary. This follows the approach of [BBM00], which considers multi-user security in a public-key setting.

When applied to the real world a "user" in the sense of multi-user security is more likely to model a key than a real person. Thus whenever a real-world user agrees or generates a new key, a new cryptographic user is created in the language of multi-user security. We adopt this terminology in the subsequent rationale and security proof. We are primarily concerned with systems where each user is able to process data up to some fixed throughput limit and for a fixed amount of time and we wish to protect against any one of these keys being compromised.

For this reason, to model multi-user security we prefer the approach of [BHT18] (but in the standard model), modelling each user as having an independent query limit that represents the amount of data they are able to process, instead of an overall global query limit for the entire system as introduced by [BCK96]. Some examples of how per-key limits might arise are, due to limits on key lifespan, bandwidth limitations or a restriction enforced by the cryptographic protocol.

2.1.2 Beyond-birthday security

We say that a mode has *birthday* security in parameter q if the advantage of adversaries against the mode is bounded by $O(q^2/2^n)$, where n is the block size of the underlying block cipher. If it has a security bound that is better (that is, smaller) than birthday security, then we say the mode has *beyond-birthday* security.

Following [BDJR97, Rog04b] we favour concrete security bounds that can be directly compared to our risk tolerance, rather than asymptotic bounds. NIST Special Publication NIST SP 800-38B

[Dwo05], highlights that adversaries may be willing to mount attacks even if their overall probability of success is very small, and so our security results should have a significant safety margin. Since security proofs for modes are often tight (in the sense that attacks often exist which almost match the proven advantage bounds) we require conservative estimates for usage levels to ensure the long-term security of a mode in this multi-user setting. In practice, most block ciphers have a block size of $n = 128$ bits which means that modes with birthday security do not have a suitable advantage bound for realistic high data cases. As a result, we require beyond-birthday security bounds.

Let us consider a hypothetical but realistic example system. It has K independent keys (users), each capable of processing q packets of length ℓ blocks in their lifetime. Suppose the adversary can perform t block cipher computations. A typical mode with birthday security in q and ℓ might have an advantage bound of approximately $\frac{(q\ell)^2}{2^n} + 2^{t-k}$, where the block size n is almost certainly 128 and k is the size of the block cipher's key.[2] By applying standard hybrid arguments, we obtain a multi-user advantage bound of

$$K \cdot \left(\frac{(q\ell)^2}{2^n} + 2^{t-k} \right).$$

The first term will become large once there are many messages or users. For a concrete example, taking $K \approx 2^{20}$ users, each processing $q \approx 2^{40}$ packets of length at most $\ell \approx 2^{16}$ blocks, the above bound is approximately $2^4 + 2^{20+t-k}$. This is greater than 1 and so gives no security assurance at all. In this example a mode with beyond-birthday security in q might have a multi-user advantage bound of

$$K \cdot \left(\frac{q\ell^2}{2^n} + 2^{t-k} \right) \approx 2^{-36} + 2^{20+t-k},$$

which might well be considered acceptable for suitable key lengths.

2.1.3 Robustness to nonce misuse via MRAE security

Legacy modes such as CBC typically require an Initialisation Vector (IV) that is unique and unpredictable. However, experience has long shown that despite this requirement, protocol engineers are sometimes seen to construct IVs from (for instance) the concatenation of known values, leading to attacks such as BEAST [DR11], which affected SSL v3.0 and TLS v1.0. These attacks are serious, having led to practical plaintext recovery attacks, and strongly suggest that where possible, requiring an unpredictable IV is best avoided. In this paper, we call an IV that is required to be unique but not unpredictable a nonce.

Even maintaining uniqueness of nonces can be hard, and there are examples of this being exploited in practice; for instance, tricking a victim into repeating a nonce was the basis for the KRACK attack [VP17] affecting almost all WPA2 implementations, which allowed an attacker to decrypt, replay, and sometimes spoof arbitrary packets sent over WiFi.

Nonce reuse breaches the security assumptions of a nonce-based mode, generally causing the loss of confidentiality and/or authenticity for the messages. For GCM, the situation is particularly severe, as Joux's "Forbidden attack" [Jou06] shows how an attacker can recover the authentication key itself, allowing them to forge arbitrary future messages.

We believe the most appropriate modern notion for nonce-misuse resistance is Misuse-Resistant AE (MRAE) security [RS06]. MRAE security describes a notion of the "best possible" nonce-based security, requiring that if two encryptions are processed under the same nonce, the only thing that an attacker can learn is whether the message and AD was identical between the two encryptions. Since every bit of ciphertext must therefore necessarily depend on every bit of input, MRAE security necessarily requires a two-pass encryption routine[3]. There have been several attempts to find an alternative to the MRAE security notion to something that can be met by online (i.e. one pass) encryption schemes (for an attempted classification, see for example [HRRV15, Fig. 11]). As [HRRV15] argues, the resulting Online AE (OAE) security notions mostly offer weaker security guarantees than would be obtained from using an MRAE scheme, making an MRAE scheme more suitable for our use cases.

[2] The 2^{t-k} term accounts for an adversary's ability to attack the block cipher directly; care is required here as this term is intuitive but unrealistic [BL12], and is provided for illustrative purposes only.

[3] But not necessarily decryption [Min20].

2.1.4 Security against release of unverified plaintext, in a strong sense

The release of unverified plaintext (RUP) occurs when an implementation of the decryption function of an AEAD scheme leaks information in the event of a decryption failure. Situations where this could occur include when the candidate plaintext is created separately from the authentication check (as in classic composition schemes like CCM or GCM) and immediately released to the user, or the candidate plaintext is stored in an intermediate buffer that is not flushed, or the candidate plaintext is written to an external buffer that other processes read before the cryptographic implementation later clears it, or protocol designers choose not to transmit the tag in order to save bandwidth (in modes where this does not prevent decryption). Such a mistake underpinned 2018's EFAIL [PDM+18] attack on OpenPGP and S/MIME when attackers were able to modify ciphertexts in a way that candidate plaintext contained malicious code that was then read by follow-on processing, leading to a total loss of confidentiality. A more subtle example is Lucky 13 [AP13], which affected TLS implementations that performed a padding check on CBC-decrypted candidate plaintexts before cryptographically authenticating them. Information about the result of the padding checks leaked via timing differences, and this allowed a machine-in-the-middle attacker to recover plaintext.

There have been many attempts to define a RUP security target and they mostly differ on the information that is leaked by a decryption failure. There is no single correct way to define this leakage because the most suitable model depends on the nature of the AEAD mode, the implementation and the user's risk appetite. For example, AEZ [HKR15] was shown to have RUP security in a model where on a decryption failure the putative plaintext is leaked. However, the same version of AEZ was proven to be insecure in an alternative RUP model if a variable-width buffer is not correctly cleared on decryption failure [BPS15, Appendix E]. Both models are valid and an implementation might reasonably guard against one or both such instance of leakage; we describe the results here to illustrate how reasonable models of leakage can lead to very different results.

We believe that a scheme labelled as RUP secure should prevent an attacker from learning any information about previously processed inputs; such abilities are core to the EFAIL attack. Some RUP security models, such as PA1 [ABL+14] and AE-RUP [CDD+19], do not prevent an attacker learning about previously encrypted inputs – indeed, GCM is PA1-secure but would not prevent the EFAIL attack. We therefore view PA1 and AE-RUP as insufficiently strong for our requirements.

The SAE notion [BPS15] generalises several models [BDPS14, ABL+14, HKR15] and aims to describe the "best possible" for a leaking implementation: leakage does not reveal any information about the key or plaintext. However, while we are satisfied it achieves this goal, SAE security still does not necessarily prevent an attacker from exerting control over the candidate plaintext. For instance, an encrypt-then-MAC scheme that decrypts ciphertext in place and on a decryption failure fails to clear the ciphertext/plaintext buffer can be proved SAE secure, but does not prevent an attacker from injecting potentially malicious data into a system. The ability of an attacker to exert control over candidate plaintexts was also key to the EFAIL attack, and blocking this ability is also a high priority of our RUP requirement. For this reason, we also view SAE security as an insufficiently strong notion.

For our security target, we favour the extended RUPAE definition of [ADL17][4], which requires that leakage is indistinguishable from random data of the correct length[5]. This security target addresses our concerns about the previous notions outlined above.

In a related but weaker notion, we require that the tag is used as an input to the decryption routine, in order to prevent users from deliberately omitting the authentication step. Note that this weaker property follows as a consequence from extended RUPAE security, since anything other than the correct tag in the decryption process will result in output that is indistinguishable from random data.

2.1.5 Block cipher-agnostic

It is important that the mode is suitable for a range of block ciphers, including AES. We made this case as 'Compatibility' in the introduction of Section 2.

[4] Though we drop the nonce restrictions.

[5] The RAE security notion of [HKR15], when used with a simulator that outputs random messages as in the proof of AEZ, contains an equivalent notion of RUP, although their definition accounts for the possibility of a variable amount of stretch.

2.1.6 Have a simple security proof

Errors in published security proofs can have serious or catastrophic consequences (e.g. [CHJS13, IIMP20]). Further, when proofs are correct, sometimes they use an inappropriate security model, make unjustified assumptions, or are too complex for the wider community to reasonably verify (for a summary of issues, see e.g. [KM19]). We therefore require a mode that offers sufficient security and robustness while also offering a security proof that can be verified as easily as possible.

Much progress has been made towards formal verification of security proofs [BGHZ11, BLS20, PM15, Bla06, Ber13, BGJZ07, AHR+21]. Such methods would add strong mitigation against errors in security proofs, and we encourage research into verification of proofs in the development of new standards for modes.

2.2 Medium priority requirements

2.2.1 Avoid ideal cipher assumptions

Some modes that simulate independent block ciphers by generating multiple keys (such as AES-GCM-SIV [GLL17]) are proven secure in the ideal cipher model. For some block ciphers, we believe that the ideal cipher model is a poor model of reality; for example related-key attacks on AES are well known (e.g. [BKN09]). We believe that reducing to a single-user computational ±PRP distinguishing game[6] where possible is preferable because it is a more realistic model and provides a more widely applicable result. Results in the ideal cipher model also do not obviously apply to quantum adversaries (see Section 2.2.2).

2.2.2 Security against quantum adversaries

When deploying new cryptographic algorithms intended to be secure for many years to come one must anticipate future threats, including the possibility of an adversary with a cryptographically-relevant quantum computer. Specifically, where modes have a classical proof of security in the standard model, we would adopt the Q1 model (as notated in [KLLN16] and introduced in [Zha12]). Where modes have a classical proof of security in the ideal cipher model, we would adopt the QROM model (as introduced in [BDF+11]). In these models, the adversary has access to their own quantum computer for offline computation, but can only interact with the honest users (and thus keys) through classical interfaces.

Concrete, efficient, black-box security reductions in the standard model translate naturally to the Q1 model since such reductions can be directly applied to quantum adversaries. The only required change to the overall security argument is that the heuristic assumption on computational ±PRP security of the underlying block cipher must now be made with respect to adversaries that perform quantum (as opposed to classical) offline computation, with the reduction demonstrated in the main security proof requiring little modification. Thus, when proofs are made in the standard model, we have a measure of confidence in the quantum Q1 security of a scheme. By contrast, classical security in the ideal cipher model does not so easily translate to security in the QROM.

Authors have suggested alternative security models to the Q1 and QROM models. For example, [DFNS11, GHS16] suggest a "frozen smart-card" attack in which the adversary is given the ability to make encryption queries in superposition. We believe that such adversarial models are unrealistic for algorithms implemented in classical hardware.

2.2.3 Performance and circuit size

In our use cases, we find that computational performance requirements can often be treated as secondary to security. Fundamentally, as long as a mode is sufficiently performant to be useful, we are prepared to trade additional efficiencies for increased security, where security might be in terms of any of the criteria listed in this section. This means we are content to make design decisions that add security at some cost in performance – MRAE security is a prime example, since it necessitates two passes over the data, and thus a buffer.

That said, we do not ignore performance, and have paid attention to implementation considerations both in hardware and software through the design process for our proposed designs. For example, although MRAE security requires two passes over the data, almost all block cipher calls

[6] For a full definition, see equation (3) of Section 4.1.

within each pass can be parallelised in hardware. When performing symmetric encryption on general-purpose computing devices such as CPUs, we note that hardware acceleration of AES via the AES-NI instruction set, and of multiplication in the Galois field with 2^n elements via PCLMULQDQ carryless multiplication instructions, usually enables hardware acceleration of designs based on these operations.

2.2.4 Context commitment and discovery

Committing encryption [GH03] was not seriously considered in the design of AEAD schemes until 2017 [GLR17]. The core idea is that a ciphertext and tag should effectively be a commitment to the "context" (i.e. the key, nonce, associated data, and message) that produced them. Context discovery is a similar notion: if context commitment security of an AEAD scheme is analogous to collision resistance of a hash function, then context discovery security is analogous to preimage resistance [MLGR23]. A scheme achieving both might be considered closer to a universal cryptogram, securely encapsulating all information, but currently standard AEAD security definitions offer no such guarantee.

Real-world security vulnerabilities have been discovered caused by a lack of context commitment security, such as the "invisible salamanders" vulnerability in Facebook's message franking scheme [DGRW18]; a vulnerability in Subscribe with Google allowing subscribers to deliver content that decrypts differently depending on which authorizer is used [ADG+22]; or a vulnerability in Shadowsocks proxy servers that allows users to trial-authenticate a few thousand passwords at the same time by submitting a single ciphertext query [LGR21]. Attacks tend to fall into two categories: scenarios where the same ciphertext is sent to more than one user and the implicit assumption that the decrypted plaintext will be the same for all users is violated; and online attacks on authentication systems involving a low-entropy key are sped up by the ability to submit a ciphertext that decrypts under many keys. In the majority of cases, it is assumed that one of the parties holding the symmetric key is corrupted, or the key has low entropy.

There have been myriad definitions of commitment-based security properties that could be targeted by AEAD schemes, each subtly different and each under different names (e.g. [FOR17, GLR17, DGRW18, ADG+22, BH22]). The strongest notions [BH22, CR22, MLGR23] approximately coincide, targeting commitment to the full context of nonce, associated data, and message, and appear most appealing.

For now, we justify making context commitment and discovery a medium-priority requirement (despite a strong emphasis on robustness) on the basis that the attacks listed above are not as severe as the attacks listed in the nonce-misuse resistance and RUP sections above, and generally require one party to already be corrupted or for the key to have low entropy. Moreover, proofs of efficient constructions tend to require stronger security assumptions on the block cipher. As context commitment is an topic of much active research we anticipate that any discovery of new and more severe attacks could prompt us to change our position. We note that there is a generic transform where context commitment and discovery security is achieved by appending a hash of the context to the ciphertext, and this gives us some amount of confidence that we could adapt a design if necessary, at the cost of additional stretch.

2.3 Low priority requirements

2.3.1 Wider classes of leakage resilience, side-channel mitigations

Modes can be at risk of leaking private information via timing variation or message lengths, especially if messages are compressed. Examples include POODLE [MDK14] and Lucky 13 [AP13], and fixing timing vulnerabilities after the fact is difficult [AP16, RPS18]. We therefore require modes whose padding mechanisms do not leak information about the content of the padded data (other than its length); and that can be implemented in a manner in which timing is independent of key and plaintext – it only depends on input length.

There are other side-channels via which a mode could leak private information, such as power consumption. We believe adding mitigations (such as masking) to the underlying block cipher and/or to well-designed modes is usually possible without unworkable performance implications. For this reason, beyond mitigating RUP, we do not consider side-channel or leakage mitigations further in this paper, except to note that it is important that our mode designs do not exacerbate potential risks, or make it difficult to add mitigations later.

2.3.2 Graceful security degradation when usage limits exceeded

We previously commented that there are often attacks on modes that achieve advantage close to (or at) the security bound. These attacks can range from a loss of confidentiality on one or a small number of messages out of a potentially large collection, up to something much worse, akin to the authentication key recovery attack on GCM, the former being more tolerable than the latter. In the event that a mode processes enough data that these attacks become viable we hope that the design of the mode is such that the loss of security is more tolerable; that security degrades gracefully, rather than suddenly and totally. This is particularly important when using modes with birthday security or block ciphers with small block sizes [BL16]. However, for the use cases we are targeting we expect our users to be sufficiently risk-averse that they would always seek to remain well within provable limits and in the absence of a wider block cipher believe this issue can be mitigated by designing beyond-birthday secure modes. Therefore, we view this as a low priority requirement.

2.3.3 Support for private nonces

The CAESAR call for submissions [Ber14] considered private nonces, since sending nonces in plaintext alongside ciphertext and tag may expose metadata that would have been better kept hidden, or conversely may prevent the user from putting useful unique inputs into their nonce. A few works have looked into the problem since (e.g. [BNT19]), but it is not clear what an optimal solution would look like, or what the trade-offs would have to be. We leave this as an interesting goal to consider in future, but as no major protocols currently require it we do not prioritise it for now.

2.4 Assessment

In this section we have described a number of goals and assigned a priority to each. We expect our goals to be relevant to the wider community, although in some cases prioritised differently.

At high priority we require the mode to be beyond-birthday secure to allow the transmission of high quantities of data. It should provide MRAE and RUP security. We require the mode to have a simple, clear security proof that gives a conservative security margin even under many users. We require the mode to be compatible both with AES and with alternative block ciphers with the same block width.

At medium priority, we would like to avoid ideal cipher assumptions, and have a mode that is also secure against quantum adversaries, to provide confidence moving into the post-quantum era. Ideally, such a mode would have good performance characteristics and context commitment security. We place lower priority on other desirable properties, such as wider classes of leakage resilience, security degradation beyond the provable bound, and support for private nonces.

2.4.1 Representative candidates

To close our requirements discussion, let us briefly consider how some notable candidates compare to our key priorities. We will introduce our proposed modes for comparison, and apologise for the many promising schemes we do not consider here, or for any points of note we do not raise: this section is intended to be representative rather than exhaustive. None of the designs discussed in this section, including ours, achieve context commitment or discovery security.

GCM GCM [MV04] is widely adopted, widely standardised, and natively supports any block cipher with a 128 bit block size. While the case of variable-length nonces is rather complex, when restricted to 96-bit nonces GCM has a particularly clean security proof. In the form of AES-GCM, it is quite possibly the most-used AEAD mode. Using only an AES encrypt engine and a Galois multiplier, it can be implemented efficiently in hardware and most modern platforms comes with accelerators for one or both of these instructions. Unfortunately, GCM is fragile, since nonce misuse leads to a catastrophic loss of security [Jou06] and there is no RUP security [ABL$^+$14].

OCB3 OCB3 [KR11] is the third member of the OCB family. It is a highly efficient mode, widely available and simple to implement. It uses a generic block cipher and comes with a neat security proof via the ΘCB tweakable construction, and then instantiates this with an XEX construction. This gives it a clear security proof, up to the birthday bound. However it does not provide meaningful resistance to misuse.

ChaCha-Poly1305 ChaCha-Poly1305 [Ber08,Ber05,NL15] is another widely-used design, which is highly efficient (especially in software) and available in several general purpose libraries. The mode itself has a security proof in the standard model. Like GCM, ChaCha-Poly1305 is fragile against misuse, though the impact of nonce misuse is slightly less severe. For our purposes, another issue is that as an all-in-one stream cipher-based mode, it would be complicated to incorporate an alternative block cipher design.

AES-GCM-SIV AES-GCM-SIV [GLL17] trades some of the efficiencies of GCM and ChaCha-Poly1305 for increased security. Specifically, it requires two passes over the data, and thus an intermediate buffer, in exchange for MRAE security. On decryption, an intermediate buffer caches the unverified candidate plaintext until the tag can be verified. Since XOR changes to ciphertext pass through to XOR changes to the candidate plaintext, the scheme does not meet strong notions of RUP security. AES-GCM-SIV has strong beyond-birthday security guarantees when nonces are not misused [GLL17], although the proof of this is in the ideal cipher model. The mode can be used with a generic block cipher, and is available as an RFC specification.

AEZ AEZ [HKR15] meets our targets for nonce-misuse and RUP (modulo earlier discussed concerns around early-abort implementations). There is a clearly defined specification available for use, and given the security goals targeted it is extremely efficient.

While the security proof assumes a standard cipher similar to AES (but with a tweak), the proposed scheme only uses a reduced round version. This means that the mode relies on a less well studied security assumption, and that replacing the internal block cipher with an AES-equivalent would significantly slow down the mode. We would also favour fixing the tag length to avoid risk of misimplementation or misuse, as the design is likely to be complicated to implement, especially in hardware [HG16].

Romulus Romulus-M is a mode that was a finalist in the NIST lightweight cryptography competition [IKMP20]. It offers beyond-birthday security, nonce-misuse resistance, and promising performance characteristics. It achieves partial RUP security through PA1 + INT-RUP. Romulus-T has many of the same advantages as Romulus-M, and adds improved leakage resilience.

Both of these schemes are built around the SKINNY tweakable block cipher, which would be complicated to replace with a generic (non-tweakable) block cipher. Furthermore, the security proof is in the ideal cipher model [GIK$^+$].

2.4.2 Our proposals: GLEVIAN and VIGORNIAN

In this paper we introduce GLEVIAN and VIGORNIAN, designed to satisfy our high-priority requirements. Both have native nonce-misuse resistance and RUP security, with a proof in the standard model which we hope is easy to verify. Both are compatible with any modern block cipher, including AES (VIGORNIAN also requires an implementation of Galois multiplication). Our mode achieves beyond-birthday security when not misused, even when supporting many users.

We have made trade-offs in the design of GLEVIAN and VIGORNIAN, the most obvious of which is in performance. Each mode can be computed in two passes and requires one internal buffer to store the input (note that we show that this does not affect our RUP security). While computations within each pass can be parallelised, GLEVIAN requires approximately three block cipher calls per block of input and VIGORNIAN requires two Galois multiplication operations and one block cipher call. Moreover, the overhead of the KDF is significant for short messages. This means that GLEVIAN and VIGORNIAN require more internal primitive calls than the other schemes mentioned in this comparison.

3 GLEVIAN and VIGORNIAN

3.1 Description of GLEVIAN and VIGORNIAN

We present a schematic of GLEVIAN and VIGORNIAN encryption routines in (respectively) Figures 1 and 2. We expect that the diagrams are largely self-explanatory, but in Section 4.1.6 we provide a key to the symbols and conventions used. In the remainder of this section we provide some explanatory detail for the diagrams in Figures 1 and 2, while in Section 3.2 we explain the choices made in the design of the modes.

GLEVIAN and VIGORNIAN are instantiated with an n-bit block cipher $\mathcal{E}$ with a k-bit key. The user's secret key is denoted by K in Figures 1 and 2; it is used to key an instance of $\mathcal{E}$ which we denote $\mathcal{E}_{(K)}$.

The encryption routines take as input a message M and associated data A, both of arbitrary length, and a nonce whose length is (slightly) less than n bits. The outputs are a ciphertext C which is the same length as M, and an n-bit tag T. Figures 1 and 2 show A, M and C as split into n-bit blocks, including a final partial block as an example (although this may not exist).

The nonce N is used to produce per-nonce k-bit keys K_{lrw2}, K_{ctr}, and either an n-bit key $K_{\mathcal{H}_1}$ and a k-bit $K_{\mathcal{H}_2}$ (in GLEVIAN); or an n-bit key $K_{\mathcal{H}}$ in VIGORNIAN. These keys, excepting $K_{\mathcal{H}_1}$ and $K_{\mathcal{H}}$, are used to key block ciphers $\mathcal{E}_{(K_{lrw2})}, \mathcal{E}_{(K_{ctr})}$ and, in GLEVIAN, $\mathcal{E}_{\left(K_{\mathcal{H}_2}\right)}$. The per-nonce keys are the outputs of the dash-boxed KDF function where the nonce is concatenated with a sequence of counter values, which are then encrypted with $\mathcal{E}_{(K)}$ as shown in the diagram; this component is described further in Section 4.2.7. This component is the source of our requirement on the block size of $\mathcal{E}$ - the block size must be large enough to encode the required counter values.

Encryption proceeds as follows. First, associated data A, plaintext M, and the n-bit all-zeroes string 0^n are processed by the LRW2 construction; that is, a keyed hash is applied to A and M to produce an intermediate value which is then used in an XOR-encrypt-XOR encryption of 0^n. For each of A and M, if its final block is partial then the keyed hash pads them using $100\cdots0$ padding; if no partial block exists then no padding occurs. The n-bit output of this LRW2 encryption is denoted I. The hash constructions in GLEVIAN and VIGORNIAN, and their padding schemes, are described in Sections 4.2.2 and 4.2.3, and the choice of hash construction is the only way in which the two modes differ. The LRW2 construction is described in Section 4.2.4. This intermediate value I is used as the IV for CTR mode encryption of the plaintext, producing ciphertext C; CTR mode is described more fully in Section 4.2.1. The AD A, ciphertext C and intermediate value I are then processed by the same LRW2 construction to produce the tag T. We do not describe decryption, which is performed in the obvious manner. Decryption failure occurs if the LRW2 decryption of I does not equal 0^n.

We note for both GLEVIAN and VIGORNIAN that encryption and decryption can be performed in two passes over the data. The first pass computes I, and the second pass produces ciphertext and LRW2-encrypts I to produce the tag. Each pass can be almost entirely parallelised.

3.2 Design decisions

In this section we offer a brief justification and overview of our design choices in GLEVIAN and VIGORNIAN.

As discussed in Section 2, we desire multi-user beyond-birthday MRAE security and (under reasonable leakage models) RUP security. We add beyond-birthday security via nonce-based key derivation, as introduced by AES-GCM-SIV, to produce a per-nonce key for an underlying birthday-secure scheme (in fact, we produce several keys to provide cryptographic separation between certain block ciphers). Having used the nonce to produce a key for an underlying mode, there is no need to further process the nonce; it suffices then to wrap a DAE[7] scheme rather than AEAD scheme with nonce-based key derivation. We select the XORP construction [Iwa06, IMV16] as a KDF because it is efficient, is well studied with multiple independent proofs of beyond-birthday security [Iwa06, Pat10] and - as it can be instantiated with the same block cipher as the rest of the mode - it obviates the need for extra computational assumptions. The resulting AEAD mode is then beyond-birthday secure when used and parameterised correctly because it encrypts only a small number of messages on any one nonce, and because attackers can learn little information from rejected decryption queries. We do not achieve fully beyond-birthday security under serious misuse or in the presence of RUP. Using nonce-based key derivation has a security cost; it drastically increases the number of keyed instances of the underlying

[7] DAE [RS06] is similar to AEAD but does not feature a nonce input.

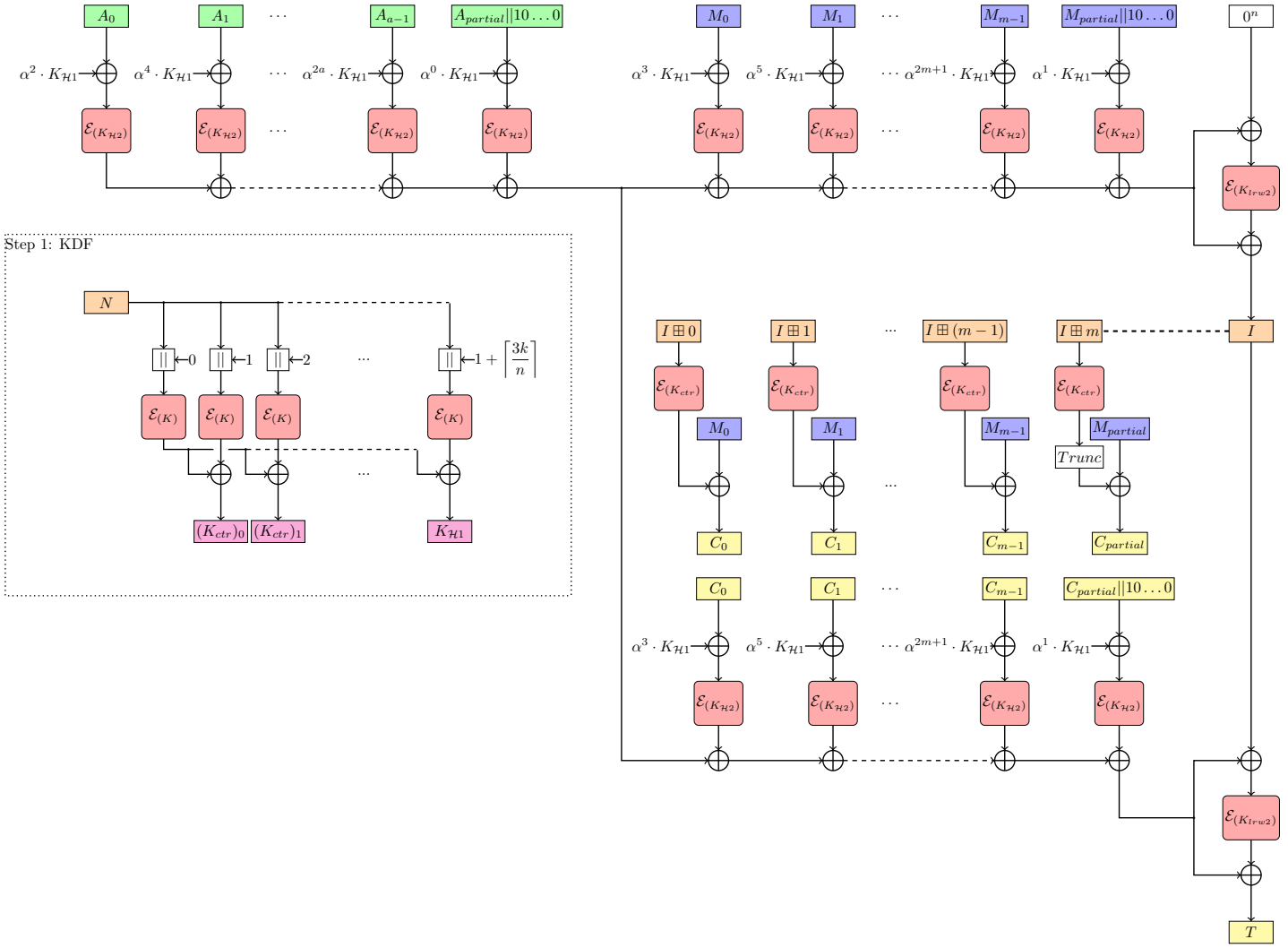


Figure 1: The full schematic of the encryption routine of GLEVIAN AEAD mode. All the wires carry n bits, apart from in three places: any partial final block of plaintext is not padded and produces a partial final block of ciphertext; the final block of associated data may be partial; and the nonce in the KDF is $n - \lceil \log\left(2 + \lceil \frac{3k}{n} \rceil\right) \rceil$ bits, with the low bits of the n-bit word reserved for concatenation with a counter. The counter in the KDF steps enough times to produce three k-bit keys and one n-bit key, which imposes a condition on the block size that $n \geq \lceil \log\left(2 + \lceil \frac{3k}{n} \rceil\right) \rceil$ - we remark that this will be satisfied by any realistic block cipher. α is a primitive element of $GF(2^n)$ and when we write $\alpha \cdot K_{\mathcal{H}_1}$ we mean to multiply these values as elements of $GF(2^n)$. Multiplication by α corresponds to stepping an LFSR; for more detail, see Section 4.2.2.

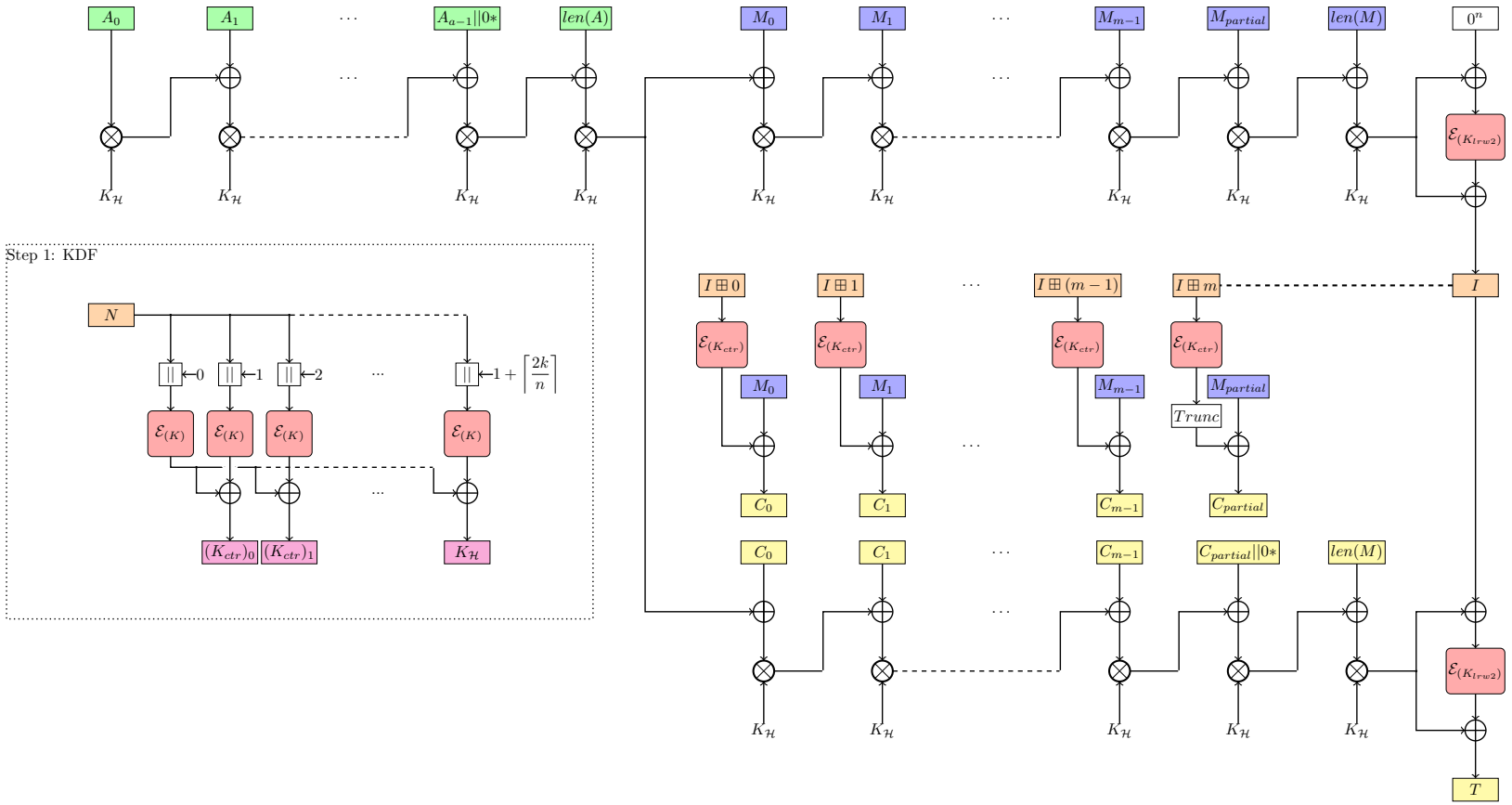


Figure 2: The full schematic of the encryption routine of GLEVIAN AEAD mode. All the wires carry n bits, apart from in three places: any final partial block of plaintext produces a partial final block of ciphertext; the final block of associated data may be partial; and the nonce in the KDF is $n - \lceil \log\left(2 + \lceil \frac{2k}{n} \rceil\right) \rceil$ bits, with the low bits of the n-bit word reserved for concatenation with a counter. The counter in the KDF steps enough times to produce two k-bit keys and one n-bit key, which imposes a condition on the block size that $n \geq \lceil \log(2 + \lceil \frac{2k}{n} \rceil) \rceil$ - we remark that this will be satisfied by any realistic block cipher. We write $\otimes$ for multiplication in $GF(2^n)$; for more detail, see Section 4.2.3.

DAE scheme as each long-term key can spawn potentially many nonce-dependent short-term keys. In our security proofs we account for this increase in the number of keyed instances of DAE schemes and show that the mode remains secure for high data uses under a reasonable assumption on the instantiating block cipher (as long as this block cipher has a sufficient size of key). Nonce-based key derivation also comes with a performance cost, especially substantial on short messages.

Our choice of DAE scheme must offer both DAE security and RUP security under reasonable leakage models, but only needs to attain birthday security in these models. We take a modular approach to defining a DAE scheme, which we present in a top-down fashion here; each component inherits security from its subcomponents. (In Section 4.2 we take a bottom-up approach and prove each component in turn to be secure, from which the security of the entire DAE mode will follow.). We use the Pad-then-Encrypt (PTE1) construction [RS06] which is a natural way to provide secure DAE and must be instantiated with a secure wide-tweak wide-block cipher. We instantiate this wide-tweak wide-block cipher with the Protected IV (PIV) construction [ST13], whose modular proof structure aligns well with our modular approach. The PIV construction itself needs to be instantiated with two subcomponents: (1) a nonce-based enciphering scheme, and (2) a wide-tweak narrow-block cipher.

For (1), we choose CTR mode which has well known security and efficiency properties. We have chosen to enlarge CTR's nonce space to be the entire width of the block cipher. This results in a stronger nonce uniqueness requirement; we now require that PIV never calls CTR with a nonce that is close to any other nonce, rather than simply requiring nonces be distinct. To prove that PIV meets this stronger requirement we give a revised security proof for our instantiation of PIV with CTR in Section 4.2.5. The simultaneous modifications to CTR and PIV improve the overall security bounds for GLEVIAN and VIGORNIAN.

For (2), we choose the LRW2 construction [LRW02]. In turn, the LRW2 construction needs to be instantiated with a hash that is ε-almost XOR universal. This choice is the difference between GLEVIAN and VIGORNIAN.

For VIGORNIAN we choose a polynomial MAC heavily inspired by GCM's GHASH component. This choice of hash is motivated by three factors: firstly, we believe it to be performant on CPU; secondly, it has well-studied security properties; and thirdly, it requires a comparatively small key, which reduces the cost of the nonce-based KDF.

GLEVIAN's hash is an alternative for VIGORNIAN's hash that may be more performant on some platforms (on platforms such as FPGAs the complexity of the Galois multiply operation in VIGORNIAN's hash can dominate resource usage). GLEVIAN's hash is based on PMAC1[8] [Rog04a] and does not require a Galois multiply operation. These advantages come at a cost; GLEVIAN's hash has a larger key than that of VIGORNIAN (thus making the nonce-based KDF more expensive), and requires one block cipher call per block of input instead of one Galois multiply, which on some platforms is likely a performance hit. Since we believe that each hash may have platforms where it is the more performant, this document presents both.

We take a modular approach to the security proofs; the security of GLEVIAN and VIGORNIAN can be established by studying each component in turn. We hope that if any choices of subcomponent are found to be undesirable it should be easy to create a variant mode by replacing a subcomponent and proving that it achieves the required security properties.

GLEVIAN and VIGORNIAN are both defined in terms of an unspecified block cipher of any size of key and essentially any block size. However, one must be aware that, because of the nonce-based key derivation step, GLEVIAN and VIGORNIAN do not achieve k bits of security from a block cipher with a k-bit key. We are able to bound this loss of security in the detailed version of our main theorems (Section 4.4) by giving results relative to an assumption on the $\pm$PRP advantage of the underlying block cipher. In particular, the security of GLEVIAN and VIGORNIAN does not rely on the ideal cipher model.

3.3 Security results

In Section 4, we will prove some formal security theorems for GLEVIAN and VIGORNIAN, which are our key results. To motivate our work, here we give simplified versions of them without proof, and provide a brief discussion on their impact. In our two RUP theorems, we assume that any variable length buffers leak, which has no impact on encryption (since the only variable length buffer is the ciphertext) and for decryption means that invalid queries leak the putative plaintext.

[8] We remove the final block cipher call of PMAC1 to use just the ε-almost XOR universal subroutine

For each result, let us assume we have an adversary with access to K independent users of GLEVIAN or VIGORNIAN. To each user, they can make up to q encryption queries on each of R distinct nonces, and f decryption (and, if applicable, leakage) queries under no constraint on nonces, where each query is of length at most ℓ n-bit blocks, where length counts the total length of AD and message or AD and ciphertext. We presume $\ell \geq 1$; that is, the adversary is not restricted to only making empty queries.

For now, we assume the modes are instantiated with a secure block cipher with a key of size $k \leq 2n$. We require that the adversary does not have sufficient computational power to meaningfully threaten the $\pm$PRP security of this block cipher[9], and we assume the blocksize n is at least 4 bits to avoid some unrealistic corner cases in our bounding.

Theorem 31 (Simplified). Suppose GLEVIAN is instantiated with a perfectly secure block cipher, and the implementation leaks variable length internal buffers. Then, the multi-user MRAE security of GLEVIAN in the RUP model with nonce-misuse, μMRAE-RUP(GVN), satisfies

$$\mu\mathsf{MRAE\text{-}RUP}(\mathrm{GVN}) \leq \frac{1}{2^{2n-11}}K(R+2f)^3 + \frac{1}{2^{n-5}}K(\ell+1)^2(R+2f)(q+2f)^2.$$

Theorem 32 (Simplified). Suppose GLEVIAN is instantiated with a perfectly secure block cipher. Then, the multi-user MRAE security of GLEVIAN with nonce-misuse, μMRAE(GVN), satisfies

$$\mu\mathsf{MRAE}(\mathrm{GVN}) \leq \frac{1}{2^{2n-11}}K(R+2f)^3 + \frac{1}{2^{n-5}}K(\ell+5)^2(q+2)\big(R(q+1)+f\big).$$

Theorem 33 (Simplified). Suppose VIGORNIAN is instantiated with a perfectly secure block cipher, and the implementation leaks variable length internal buffers. Then the multi-user MRAE security of VIGORNIAN in the RUP model with nonce-misuse, μMRAE-RUP(VGN), satisfies

$$\mu\mathsf{MRAE\text{-}RUP}(\mathrm{VGN}) \leq \frac{1}{2^{2n-10}}K(R+2f)^3 + \frac{1}{2^{n-1}}K(\ell+6)^2(R+2f)(q+2f)^2.$$

Theorem 34 (Simplified). Suppose VIGORNIAN is instantiated with a perfectly secure block cipher. Then, the multi-user MRAE security of VIGORNIAN with nonce-misuse, μMRAE(VGN), satisfies

$$\mu\mathsf{MRAE}(\mathrm{VGN}) \leq \frac{1}{2^{2n-10}}K(R+2f)^3 + \frac{1}{2^{n-1}}KR(\ell+11)^2(q+1)^2 + \frac{1}{2^{n-6}}K(q+1)(\ell+2)f.$$

3.3.1 Discussion

At first the presence of $q^2/2^n$ terms in our security bounds would seem to contradict our claims of beyond-birthday security. However, q counts the number of encryption queries *on any one nonce*, with the number of nonces given by R. Therefore when GLEVIAN and VIGORNIAN are used correctly, $q = 1$, and our bounds provide meaningful security guarantees even as R exceeds $2^{n/2}$ as required for beyond birthday security. The q^2 terms merely describe how gracefully the security decays with nonce misuse.

We account for decryption queries separately with f. Here, the advantage bounds without RUP provide meaningful security even as f (and $R + f$) approach $2^{2n/3}$ - that is, we achieve beyond birthday security in this case. In the presence of RUP our bounds are not beyond-birthday, being $O(f^3/2^n)$. We discuss this in Section 4.2.8.

Our full theorems (Section 4.4) also contain unwieldy expressions for block cipher security in terms of the $\pm$PRP advantage of multiple adversaries. Ordinarily one might bound these by the maximum advantage over all adversaries satisfying some computational constraint, but such bounds are known to be surprisingly loose (demonstrated in [BL13]). We circumvent this looseness by ensuring that our adversaries are always efficiently constructible, which we discuss further in Section 4.1.5. The result is that our block cipher security terms can be more reasonably assumed to be negligible, at the cost of less concise theorem statements.

[9]We will see in our main theorems, where we cost exactly a reduction to the $\pm$PRP security of the block cipher, that this requires the adversary's computational power be bounded by t where $t \ll 2^k/K(Rq+2f)$, and also that $K(Rq+2f) \ll 2^{k/2}$.

3.3.2 A representative instantiation

To see what this looks like in practice, consider the setting in Section 2.1.2. Here we have $K \approx 2^{20}$ users, each processing approximately 2^{40} packets (meaning $Rq \leq 2^{40}$) of length at most $\ell \approx 2^{16}$ blocks. We can instantiate our mode with a secure block cipher with $n = 128$ block size, and sufficiently large keyspace. We shall now consider the advantage bounds for GLEVIAN in various scenarios; the bounds for VIGORNIAN are similar but slightly lower (stronger), so that the GLEVIAN results hold there too.

First we consider the case where the mode is used correctly: nonces are not repeated in encryption queries (that is, $q = 1$) and unverified plaintext is not released. We allow the number of forgery attempts f to match the number of encryption queries, $Rq = 2^{40}$. Using the simplified version of Theorem 32:

$$\mu\mathsf{MRAE}(\mathrm{GVN}) \leq 2^{-27.8}.$$

This is a very low advantage which allows us to enjoy substantial confidence in the mode's security when used correctly.

Next, we consider the degradation of security when nonce repeats occur in encryption, while still disallowing the release of unverified plaintext. For example, suppose each nonce is repeated $q = 2^8$ times (but the total number of encryption queries Rq remains fixed). The advantage bound rises linearly with q:

$$\mu\mathsf{MRAE}(\mathrm{GVN}) \leq 2^{-21.0}.$$

This is still low, which illustrates considerable robustness to nonce misuse.

We now turn to the case that unverified plaintexts are released, using the simplified version of Theorem 31. This bound is cubic in the number of RUP forgeries, so we do not expect it to offer reassurance in the case that all 2^{40} forgery attempts have their unverified plaintext released. However, our primary question is how gracefully the security degrades.

For example, suppose we allow 2^8 RUP forgeries and 2^8 repeats of each nonce. Then,

$$\mu\mathsf{MRAE\text{-}RUP}(\mathrm{GVN}) \leq 2^{-19.8}.$$

This is still low, which illustrates graceful degradation of security in this scenario.

Now suppose we allow 2^8 RUP forgeries but disallow nonce repeats (and thus correspondingly increase the permitted number of nonces, since we are holding Rq constant). Then,

$$\mu\mathsf{MRAE\text{-}RUP}(\mathrm{GVN}) \leq 2^{-13.0}.$$

This bound is actually higher (weaker) than in the case of nonce repeats. This seemingly paradoxical behaviour arises from our modelling and may or may not reflect reality. Nonetheless, we can still be sure of retaining a meaningful degree of security in the event of a moderate amount of RUP.

3.4 Future work

We leave several avenues unexplored with this mode. To further our hope that others might be able to improve on our design, we list some possible improvements here.

Alternative block ciphers We are constrained by the block size of AES and its lack of tweak input. If we were able to depart from our requirement for compatibility with AES (and alternatives), we might be able to improve the design. Designing for a block cipher with block size 256 rather than 128 would allow us to achieve our desired data rates with a birthday-bound mode, without need for the nonce-based key derivation construction and simplify the design and security proof significantly. Designing for a block cipher with a tweak input might allow us to modify the design in various ways to make it more efficient. Designing modes for a block cipher with a tweakey input [JNP14] would offer even further flexibility, although perhaps at the cost of meeting the requirements in Sections 2.2.1 and 2.2.2 to a lesser extent.

Improvements in security bound While we achieve beyond-birthday security in the number of encryption and decryption queries, our security bounds are still quadratic (i.e. birthday) in the maximum message length ℓ. A design that was beyond-birthday in ℓ would allow for significantly higher usage limits.

Our beyond-birthday security bound only applies when the mode is not misused. When nonces repeat, the security of the mode degrades to birthday, and there are obvious attacks to demonstrate this bound is tight. Further, when RUP occurs the bound becomes worse than birthday, although we know of no attack that matches the bound. Attempting to find a proof that sharpens the RUP bound, or alternatively an attack that meets the bound might be an interesting avenue of future research.

Performance We do not offer significant optimisation advice in the description of our design. Comprehensive performance comparisons (especially on FPGA) might reveal where our design has strong or weak performance. We have not specified exactly how to implement VIGORNIAN's polynomial hash, and incorporating the ideas from the POLYVAL [GLL17] improvement to GHASH would likely increase performance. Our performance on short messages is probably weaker than competing designs; this may be an area in which can be improved.

Alternative security models Our design currently lacks context commitment and discovery security, and a design that added these features to the properties GLEVIAN and VIGORNIAN provide could be could be very appealing, depending on how much it required compromise on other requirements. Similarly, additional leakage resilience or side channel resistance might feasibly make the modes more attractive.

4 Proofs of Security for GLEVIAN and VIGORNIAN

In this section we present our security proofs for GLEVIAN and VIGORNIAN modes. In Section 4.1 we give the necessary preliminaries for our security proof; we encourage the busy reader to read Section 4.1.2 where we describe the notation we use throughout for distinguishing advantage. In Section 4.2 we simultaneously define and give security results for the building blocks of GLEVIAN and VIGORNIAN, starting from the lowest level components and building to a generic KDF-based AEAD mode. When we assemble the security results in Section 4.2, we find that the security of the AEAD mode depends on the multi-user security of the underlying DAE scheme. We give two bounds on this multi-user security in Section 4.3: a loose bound in a permissive adversarial model, and a strong (beyond-birthday) bound in a more restrictive model. Finally, in Section 4.4, we use the building blocks of Section 4.2 and the results of Section 4.3 to give our main security theorems for GLEVIAN and VIGORNIAN.

4.1 Preliminaries

Throughout this paper, we will typically use a standard collection of letters to denote different parameters:

Letter	Meaning
R	Number of different GLEVIAN or VIGORNIAN nonces (per GLEVIAN or VIGORNIAN key)
q	Number of times each nonce is repeated (per GLEVIAN or VIGORNIAN key)
q	Alternatively, a generic restriction on the overall number of queries
f	Number of forgery or leakage attempts per key (across all nonces)
$Rq + 2f$	Total number of queries per GLEVIAN or VIGORNIAN key (implied)
K	Total number of GLEVIAN or VIGORNIAN keys
$K(Rq + 2f)$	Total number of GLEVIAN or VIGORNIAN queries (implied)
t	Number of units of work that an adversary is restricted to
n	Block size of the block cipher
k	Key size of the block cipher
ℓ	Maximum length of a query, in blocks. The length of a query is carefully defined in Section 4.1.1
s_ℓ	Computational cost of simulating a call to GLEVIAN or VIGORNIAN with inputs of length at most ℓ blocks, or the cost of a lookup into a table of size at most $R + 2f$, whichever is greater.

We will write $\Sigma = \{0, 1\}$ to be the set containing two bits and Σ^n to be the set of n-bit words. We will write Σ^* to denote arbitrary-length bitstrings, and $\Sigma^{\geq n}$ for bitstrings of length at least n. $\pi(\star, \star)$ is a *tweakable* permutation if for all *tweaks* T, $\pi(T, \star)$ is a permutation. We define and notate tweakable functions similarly. If a function $f : \Sigma^* \to \Sigma^*$ is such that for all inputs X, the length in bits of $f(X)$ is exactly m bits greater than the length of X, then we say that f is a stretch-m function. Similarly, we define a shrink-m function to be a function with stretch-$(-m)$. If a function or permutation input is drawn from Σ^n, then we call that input *narrow*; otherwise, if it is drawn from Σ^* or $\Sigma^{\geq n}$, then we call that input *wide*. It should be clear from context which of Σ^* and $\Sigma^{\geq n}$ is meant. For example (after keys have been fixed and subcomponents instantiated), an efficient block cipher is typically a narrow-block permutation, CTR mode is a narrow-tweak wide-block permutation, and the PIV construction is a wide-tweak wide-block permutation. When we are assigning the value X to the object Y, we will write $Y \leftarrow X$. When we are drawing randomly from a probability distribution S or uniformly from a set S, we will write $s \xleftarrow{\$} S$ (noting that this is a slight abuse of notation). We will denote the range and domain of a function f as $\mathrm{Ran}(f)$ and $\mathrm{Dom}(f)$. The letter F (and G when

two are required) will be used for a random function[10].

Oracles and multi-user oracles We typically use script characters such as $\mathcal{O}, \mathcal{P}$ to denote oracles. Some oracles offer any or all of an encrypt, decrypt and leakage *interface*. We sometimes use $+$ to denote an encrypt interface, so $+\mathcal{O}$ is the encrypt interface to an oracle $\mathcal{O}$. Similarly, we sometimes use $-$ to denote a decrypt interface, so $-\mathcal{O}$ is the decrypt interface to $\mathcal{O}$. We will typically use the symbol $\sim$ to denote the leakage interface to an oracle. We combine these symbols when an oracle provides access to multiple interfaces, so $\pm\mathcal{O}$ provides an encrypt and decrypt interface but no leakage interface; while $\underset{\sim}{\pm}\mathcal{O}$ provides all three of encrypt, decrypt and leakage interfaces. When the interfaces offered by $\mathcal{O}$ are clear from context, we might just write $\mathcal{O}$.

A *multi-user oracle with K users* is a collection of K independent[11] copies of an oracle. We denote a multi-user instance of $\mathcal{O}$ with K users as $K \times \mathcal{O}$. Every multi-user oracle $K \times \mathcal{O}$ can be viewed as a single-user oracle with K times as many interfaces. It will at times be convenient to take this perspective when viewing multi-user oracles; we will do so throughout this document.

Sometimes an oracle or an interface $\mathcal{O}$ might be defined in terms of some other oracle $\mathcal{P}$, meaning that when $\mathcal{O}$ is queried, it makes zero or more queries to $\mathcal{P}$ in order to respond. We denote this relationship $\mathcal{O}[\mathcal{P}]$ and refer to $\mathcal{P}$ as a *suboracle*.

Computational and information-theoretic adversaries An *adversary* is an algorithm that runs in a game. We typically denote adversaries by capital letters, e.g. A. In parts of this document, we will not restrict the runtime of our adversaries (though we will insist that they terminate). Instead, we will allow our adversaries to have unlimited computational resources. We will call these adversaries *information theoretic*. In other parts, we will wish to only consider adversaries that have a bounded runtime. In this case, we will call these adversaries *computational*. We do not impose a model of computation on these adversaries and our results are largely agnostic of computational model. Where the computational cost of a reduction is important we leave the cost symbolic. Oracle calls will cost 0 units of work.[12]

Games A *game* is an algorithm taking as input an adversary A and an oracle $\mathcal{O}$. Typically, the output of a game is a bit. Games are frequently used to define security notions, as we will see in the next definitions. The distinguishing game G_{dist} is defined as follows.

Algorithm 1 Game G_{dist}, on input oracle $\mathcal{O}$ and adversary A

1: Initialise $\mathcal{O}$
2: A makes queries to the oracles of $\mathcal{O}$ adaptively, in any order
3: $b \leftarrow A$
4: **return** b

Since A always outputs a bit, G_{dist} also always outputs a bit. Since $\mathcal{O}$ is randomised, we will write the output of G_{dist} as a random variable that takes bit values, and denote it as $G_{\mathsf{dist}}^{A,\mathcal{O}}$. In fact, where it is clear from context that the game in question is G_{dist}, we may even abbreviate this notation to $A^{\mathcal{O}}$. We can write the probability that the game outputs 1 as $\mathbb{P}\left[G_{\mathsf{dist}}^{A,\mathcal{O}} \Rightarrow 1\right]$, or $\mathbb{P}\left[A^{\mathcal{O}} \Rightarrow 1\right]$ for short. Throughout this paper we assume A has no access to the state of $\mathcal{O}$, and may only interact with it via the game, by making queries.

Lazy sampling At various points, we shall deal with games with *lazily* and *eagerly* sampling oracles. An oracle is lazily sampled if its output values are determined when and only when they are needed to answer an oracle call. An oracle is eagerly sampled if its values are entirely determined upfront, and the game simply looks up the predetermined value at run time.

[10] That is, for domain $\mathcal{X}$ and codomain $\mathcal{Y}$, F is a uniform random deviate drawn from $\mathcal{Y}^{\mathcal{X}}$. $\mathcal{X}, \mathcal{Y}$ will typically be clear from context.

[11] By which we mean that the oracles do not share state and use independent random coins.

[12] While it may seem that oracle calls should cost 1 unit of work, making oracle calls free is a permissive change and simplifies the statements of our results.

AEAD and DAE schemes Throughout, we mostly follow the definitions established in [RS06]. We consider an *authenticated encryption with associated data* (AEAD) scheme to be a pair of maps (Π, Π^{-1}) with signatures

$$\Pi : \mathsf{K} \times \mathsf{N} \times \mathsf{A} \times \mathsf{M} \to \mathsf{C} \times \mathsf{T} \qquad \Pi^{-1} : \mathsf{K} \times \mathsf{N} \times \mathsf{A} \times \mathsf{C} \times \mathsf{T} \to \mathsf{M} \cup \{\perp\}$$

where we refer to the elements of K, N, A, M, C and T as (resp.), keys, nonces, associated data (or AD), messages, ciphertexts and tags, and where $\perp$ is the special symbol meaning reject.[13] By contrast, a *deterministic authenticated encryption* (DAE) scheme (Π, Π^{-1}) does not take a nonce input, and thus has signature

$$\Pi : \mathsf{K} \times \mathsf{A} \times \mathsf{M} \to \mathsf{C} \times \mathsf{T} \qquad \Pi^{-1} : \mathsf{K} \times \mathsf{A} \times \mathsf{C} \times \mathsf{T} \to \mathsf{M} \cup \{\perp\}$$

We write $\Pi_K(\ldots)$ to mean $\Pi(K, \ldots)$, and likewise for Π_K^{-1}. For AEAD schemes, we require that whenever $(C, T) = \Pi_K(N, A, M)$ we have $\Pi_K^{-1}(N, A, C, T) = M$, and for DAE schemes we require similarly that decryption is the inverse of encryption. When we consider the leakage interface $\sim\Pi$ of an AEAD scheme, it has signature

$$\sim\Pi : \mathsf{K} \times \mathsf{N} \times \mathsf{A} \times \mathsf{C} \times \mathsf{T} \to \mathsf{M} \cup \{\top\}$$

where $\top$ is the special symbol meaning accept. Whenever $\Pi_K^{-1}(N, A, C, T) \neq \perp$ we have $\sim\Pi_K(N, A, C, T) = \top$. Similarly, we define the leakage interface of a DAE scheme by removing the nonce input.

In many cases we will simply write Π or $\pm\Pi$ to denote an AEAD or DAE scheme rather than (Π, Π^{-1}), where the decryption interface Π^{-1} can be either unspecified or is clear from context.

We will express the advantage of an adversary A against a scheme as its advantage in distinguishing $\pm\Pi$ from $(F, \perp)$ where F is a tweakable random function of suitable signature and $\perp$ is the oracle that always returns the reject symbol. Looking ahead to Section 4.1.2 we introduce notation for an adversary's advantage; specifically for an AEAD scheme, for an adversary A we can then write its advantage against Π as

$$\mathbf{Adv}_A^{\begin{pmatrix} +\Pi, & -\Pi \\ F, & \perp \end{pmatrix}}. \tag{1}$$

Following convention, we name this advantage according to a few conditions. If Π is an AEAD scheme and A promises not to make two encryption queries with the same nonce, we call (1) *authenticated encryption (AE) advantage*. If Π is an AEAD scheme and A may repeat nonces, we call (1) the *misuse-resistant authenticated encryption (MRAE) advantage*. Finally, if Π is a DAE scheme, we call (1) the *deterministic authenticated encryption (DAE)* advantage. Informally, we call a scheme secure if a suitable subset of adversaries (such as those asking a limited number of queries and/or with limited computational resources) have their advantage upper bounded by some sufficiently small advantage. Our query notation (see Section 4.1.2) will make it clear whether an adversary is permitted to repeat nonces or not. In general, in the proofs we shall be assuming that an adversary may repeat nonces, apart from in arguments in Section 4.2.1 about the CTR subcomponent. Note that since DAE schemes do not take nonce inputs, we do not need a corresponding notion of misuse-resistant DAE.

When we wish to consider the robustness of an AEAD or DAE scheme to misimplementation such as release of unverified plaintext we follow [ADL17] and consider the advantage of an adversary in distinguishing $\underset{\sim}{\pm}\Pi$ from the tuple $(F, \perp, G)$ where F, G are independent random functions, $\perp$ always returns reject, and where $\sim\Pi$ is the leakage interface for Π (which will always be specified in context). The leakage oracle provides the adversary access to information that would be leaked from a decryption query that would otherwise return $\perp$. In the notation of Section 4.1.2 this is denoted

$$\mathbf{Adv}_A^{\begin{pmatrix} \underset{\sim}{\pm}\Pi \\ F, \perp, G \end{pmatrix}}$$

recalling that we write $\underset{\sim}{\pm}\Pi$ for a scheme Π that provides access to an encryption, decryption and leakage interface.

[13] Note that we frequently overload notation and also use $\perp$ to be the oracle that always returns the reject symbol; the meaning of $\perp$ should always be clear from context.

4.1.1 The length of adversary queries

When a query made by an adversary to an oracle, we calculate its length in a specific way. Except when clear from context, length is always measured as an integer number of n-bit blocks. An input with multiple terms (e.g. plaintext and associated data) is considered to have its length be the sum of the rounded-up lengths of its individual terms.

4.1.2 Syntax for distinguishing games

In this section we extend the delta notation of [ABL+14,Bar17] for the distinguishing advantage of an adversary between two oracles. For oracles $\mathcal{O}$ and $\mathcal{P}$ and an adversary A we write the distinguishing advantage of A between $\mathcal{O}$ and $\mathcal{P}$ as

$$\overset{\Delta}{A}\begin{pmatrix}\mathcal{O}\\ \mathcal{P}\end{pmatrix} = \left|\mathbb{P}\left[G^{A,\mathcal{O}}_{\mathsf{dist}} \Rightarrow 1\right] - \mathbb{P}\left[G^{A,\mathcal{P}}_{\mathsf{dist}} \Rightarrow 1\right]\right|.$$

This notation naturally extends to express the distinguishing advantage of an adversary A between a tuple of n oracles or interfaces (for instance $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n$ or $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$):

$$\overset{\Delta}{A}\begin{pmatrix}\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_n\\ \mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\end{pmatrix} = \left|\mathbb{P}\left[G^{A,\mathcal{O}_1,\mathcal{O}_2,\ldots,\mathcal{O}_n}_{\mathsf{dist}} \Rightarrow 1\right] - \mathbb{P}\left[G^{A,\mathcal{P}_1,\mathcal{P}_2,\ldots,\mathcal{P}_n}_{\mathsf{dist}} \Rightarrow 1\right]\right|. \tag{2}$$

We believe that this notation aids readability and is particularly suited to deterministic game-hopping proofs. First, traditional notation has the limitation that oracles available to A are usually made explicit only by the description of an accompanying security game. Here, all oracles can be made explicit as in (2). Furthermore, as we shall see, constraints on the number of queries made to each oracle can be made explicit without overly cluttering the notation, particularly when imposing fine-grained restrictions on the number of queries allowed to *each* oracle. Traditionally such restrictions are only included in a preamble to the given security result[14]. Our notation is thus arguably more expressive whilst still remaining intuitive and compact.

We will often seek to rule out pointless queries, which we define below:

Definition 1 (Pointless query)**.** *An adversary's query to an oracle is* pointless *if the answer to the query is completely determined by the answer to the adversary's previous queries.*

For example, an adversary makes a pointless query if they repeat an encryption or decryption query, or if they forward the output of an encryption interface directly to a decryption interface. We will begin most of our proofs by seeking to rule out pointless queries. We shall do so by making use of the following lemma:

Lemma 2 (Adversaries have no need to make pointless queries)**.** *Let A be an adversary, and $\mathcal{O}, \mathcal{P}$ be oracles. Suppose that A makes pointless queries. Then there is a derived adversary $\mathcal{R}(A)$ such that $\mathcal{R}(A)$ makes no more queries than A, does not make pointless queries, and achieves the same advantage as A - i.e.*

$$\overset{\Delta}{A}\begin{pmatrix}\mathcal{O}\\ \mathcal{P}\end{pmatrix} = \overset{\Delta}{\mathcal{R}(A)}\begin{pmatrix}\mathcal{O}\\ \mathcal{P}\end{pmatrix}$$

Proof. See Appendix A.1 □

An *ideal primitive* is a random variable that is uniform over the set of possible values it could take. We reason about two types of ideal primitive:

- A (Tweakable) Random Permutation, or **(T)RP** is a uniform random variable sampled from the set of (tweakable) permutations. We typically denote (T)RPs with Greek letters such as π or Ψ.
- A (Tweakable) Random Function, or **(T)RF** is a uniform random variable sampled from the set of (tweakable) function. We typically denote (T)RFs with Latin letters such as F or G.

[14] While our notation is capable of expressing these restrictions, we endeavour to also include them in our lemma and theorem statements to assist the reader.

Continuing with our conventions, we might describe ideal primitives as wide or narrow, or to have stretch-m or be length-preserving. In each case, our descriptions will be aimed at unambiguously describing the set of possible values that the ideal primitive can take as inputs and produce as outputs. Lazily sampled implementations of TRPs and TRFs are jointly defined in Figure 18. In the lazily sampled implementations, we extend the definition of the TRFs to have both an encrypt and a decrypt interface. We say the encrypt and decrypt interfaces of a TRF are pseudoinverses of each other (to remind the reader that a TRF with overwhelming probability does not have a true inverse).

4.1.3 Restrictions on adversaries

We will often consider adversaries that are restricted in some way. There are myriad ways in which we might restrict an adversary's query and computational powers, especially in the presence of multi-user oracles. Throughout the paper we endeavour to reserve the symbols described in Table 1 to each refer to a particular type of constraint.

Adversary Restriction	Example oracle(s)	Meaning
q	$\pm\mathcal{O}$	The adversary may not make more than q queries in total, to both interfaces of $\pm\mathcal{O}$.
q, f	$\pm\mathcal{O}$	The adversary may not make more than q queries to $+\mathcal{O}$ and f queries to each of $-\mathcal{O}$ and (if present) $\sim\mathcal{O}$.
q, f	$\underset{\sim}{\pm}\mathcal{O}$	The adversary may not make more than q queries to $+\mathcal{O}$ and f queries to each of $-\mathcal{O}, \sim\mathcal{O}$.
t	$\pm\mathcal{O}$ $\underset{\sim}{\pm}\mathcal{O}$	The adversary has computational cost upper bounded by t units of work.
ℓ	$\pm\mathcal{O}$ $\underset{\sim}{\pm}\mathcal{O}$	The adversary may not make any query of length greater than ℓ blocks. Length is defined in Section 4.1.1.
R, q, f	$\pm\mathcal{O}$ $\underset{\sim}{\pm}\mathcal{O}$	The adversary's queries to $+\mathcal{O}$ must contain at most R unique nonces, and each nonce must be repeated at most q times, for a total of at most Rq queries to $+\mathcal{O}$. The adversary may make at most f queries to each of $-\mathcal{O}$ and (if present) $\sim\mathcal{O}$. There is no restriction on the nonce used in each forgery attempt or leakage query.
$(R \times q), f$	$(R+f) \times \pm\mathcal{O}$ $(R+2f) \times \underset{\sim}{\pm}\mathcal{O}$	The example multi-user oracle consists of $R+f$ independent copies of the oracle $\pm O$. The adversary must make at most q queries to each of the first R copies of $+\mathcal{O}$, and no queries to the last f copies of $+\mathcal{O}$. The adversary may make at most f forgery attempts and (if present) leakage queries, but each of those can be to **any** of the $(R+f)$ or $(R+2f)$ decrypt interfaces $-\mathcal{O}$ or leakage interfaces $\sim\mathcal{O}$.
$K \times q$	$K \times \pm\mathcal{O}$ $K \times \underset{\sim}{\pm}\mathcal{O}$	The restriction described in the q row applies to the adversary separately in each of the K oracles.
$K \times (q, f)$	$K \times \pm\mathcal{O}$ $K \times \underset{\sim}{\pm}\mathcal{O}$	The restriction described in the q, f row applies to the adversary separately in each of the K oracles.
$K \times (R, q, f)$	$K \times \pm\mathcal{O}$ $K \times \underset{\sim}{\pm}\mathcal{O}$	The restriction described in the R, q, f row applies to the adversary separately in each of the K oracles.

Table 1: Table of notations for adversarial restrictions.

Restrictions can be combined. As an example, consider the distinguishing advantage of an adversary A between K independently keyed instances of some efficient block cipher $\mathcal{E}$ and K independent

instances of an RP π where the computational cost[15] of A is upper bounded by t and A makes at most q queries to each of the K oracles. Then we adapt our notation to write the query restrictions next to A and the computational restrictions underneath A, as in the following:

$$\underset{t}{\overset{\Delta}{A,K\times q}}\begin{pmatrix}K\times(\pm\mathcal{E})\\K\times(\pm\pi)\end{pmatrix}$$

In the presence of decryption and leakage interfaces we sometimes count queries to the decryption and leakage interfaces separately to the encryption interface, as can be seen in Table 1. This allows us to provide more nuanced security results such as Theorems 32 and 34.

To expand on Table 1, for a single-user oracle we write R, q, f to mean that an adversary's encryption queries must contain at most R unique nonces and no more than q *encryption* queries to any one nonce. The adversary also may make f queries to the oracle's *decryption* interface and f queries to the oracle's *leakage interface*, each under no nonce restriction. When encryption queries model data encrypted by honest parties and observed by an adversary, this models the control that honest users can exert over the amount of data they transmit and over the number of nonces they use. On the other hand, when decryption queries represent an attacker maliciously interacting with the system they are under no compulsion to respect nonce usage limits! We therefore draw a distinction between the restrictions on nonces for encryption and decryption queries. Furthermore, when leakage queries represent information leaked from an attacker's malicious decryption query, we model this by allowing an adversary one leakage query for each decryption query they make. This is a permissive model as the reality we are trying to model is that for each (N, A, C, T) an attacker submits they observe either plaintext or leakage, while in our model an adversary is under no compulsion to replay their decryption queries to the leakage interface. This model simplifies our notation and proofs and inflates an adversary's advantage by only a very small factor.

For a multi-user oracle with query restrictions $(R \times q), f$ we apply the same reasoning with a user replacing the role of nonce. Note that in this case we always give the adversary access to $R + f$ oracles of the form $\pm\mathcal{O}$ or (when a leakage interface is also present) $R + 2f$ oracles of the form $\underset{\sim}{\pm}\mathcal{O}$. This is the maximum number of distinct users they may ever query under their query restrictions - one could read (for instance) $(R + 2f) \times (\underset{\sim}{\pm}\mathcal{O})$ as simply meaning sufficient copies of $\underset{\sim}{\pm}\mathcal{O}$ to satisfy the adversary's query capabilities.

4.1.4 Standard results

Now we aim to reassure the reader that the syntax introduced in Section 4.1.2 is useful, by restating a collection of standard results. Proofs of each appear in Appendix A.1.

Lemma 3 (Triangle inequality; Δ has metric-like properties)**.** *For any adversary A and oracles $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$,*

$$\overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_1\\\mathcal{O}_3\end{pmatrix} \le \overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_1\\\mathcal{O}_2\end{pmatrix} + \overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_2\\\mathcal{O}_3\end{pmatrix}$$

Lemma 4 (Replacing suboracles)**.** *Let $\mathcal{O}$ be an oracle making use of a suboracle, denoted π. Suppose that a query to $\mathcal{O}$ generates at most r queries to π and suppose $\mathcal{O}$ does not access the internal state of π. Let s be the computational cost of simulating a call to $\mathcal{O}$ given oracle access to π.*

Let π_1, π_2 be two instantiations of π. Then there is a reduction $\mathcal{R}$ such that for any adversary A making at most q queries,

$$\underset{t}{\overset{\Delta}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_1]\\\mathcal{O}[\pi_2]\end{pmatrix} \le \underset{t+sq}{\overset{\Delta}{\mathcal{R}(A),rq}}\begin{pmatrix}\pi_1\\\pi_2\end{pmatrix}.$$

Corollary 5 (Replacing suboracles)**.** *Let $\mathcal{O}, \pi_1, \pi_2, r, s$ be as above, and let $\mathcal{P}$ be an oracle with the same input and output types as $\mathcal{O}$. Then there is a reduction $\mathcal{R}$ such that for any adversary A making at most q queries*

$$\underset{t}{\overset{\Delta}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_1]\\\mathcal{P}\end{pmatrix} \le \underset{t+sq}{\overset{\Delta}{\mathcal{R}(A),rq}}\begin{pmatrix}\pi_1\\\pi_2\end{pmatrix} + \underset{t}{\overset{\Delta}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_2]\\\mathcal{P}\end{pmatrix}.$$

[15] We leave the unit of work undefined; see Section 4.1.5

Lemma 6 (Fundamental lemma of game-playing [BR06]). *Let $G^{A,\mathcal{O}}$ and $H^{A,\mathcal{P}}$ be identical-until-*bad*, in the sense of [BR06]. Then*

$$|\mathbb{P}(G^{A,\mathcal{O}} \Rightarrow 1) - \mathbb{P}(H^{A,\mathcal{P}} \Rightarrow 1)| \leq \mathbb{P}[G^{A,\mathcal{O}} \textit{sets } \mathsf{bad}].$$

In particular, if $G = H = G_{\mathsf{dist}}$, and A is an adversary, then

$$\overset{\Delta}{A}\begin{pmatrix}\mathcal{O}\\ \mathcal{P}\end{pmatrix} \leq \mathbb{P}[G_{\mathsf{dist}}^{A,\mathcal{O}} \textit{sets } \mathsf{bad}].$$

Lemma 7 (Strong TPRP-TPRF switching lemma (adapted from [HR03])). *Let π be a TRP with domain Σ^n, and let F be a TRF with the same domain. Let A be an adversary that may make pointless queries. Suppose A makes queries on at most R tweaks, and on each tweak makes at most q queries, and that A makes at most f queries overall. Then*

$$\overset{\Delta}{A,q}\begin{pmatrix}\pm\pi\\ \pm F\end{pmatrix} \leq R \cdot \frac{q(q-1)}{2^{n+1}} \leq R \cdot \frac{q^2}{2^{n+1}}$$

and

$$\overset{\Delta}{A,f}\begin{pmatrix}\pm\pi\\ \pm F\end{pmatrix} \leq \frac{f(f-1)}{2^{n+1}} \leq \frac{f^2}{2^{n+1}}.$$

Lemma 8 (Multi-user advantage to single-user advantage - adapted from [BCK96, Lemma 3.2]). *Let $\mathcal{O}, \mathcal{P}$ be oracles. Let A be an adversary making at most $R \times q$ queries to the multi-user oracles $R \times \mathcal{O}$ or $R \times \mathcal{P}$. Let s be the maximum computational cost of simulating either of $\mathcal{O}$ or $\mathcal{P}$. Then there is an explicit reduction $\mathcal{R}$ making at most q queries and with computational cost $t + (R-1)qs$ such that*

$$\overset{\Delta}{\underset{t}{A,R\times q}}\begin{pmatrix}R\times\mathcal{O}\\ R\times\mathcal{P}\end{pmatrix} = R \cdot \overset{\Delta}{\underset{t+(R-1)qs}{\mathcal{R}(A),q}}\begin{pmatrix}\mathcal{O}\\ \mathcal{P}\end{pmatrix}.$$

Multi-user versions of the above results Each of these results can be adapted to the multi-user setting, by viewing the multiple users' oracles as one overall oracle to which each query specifies a user and an input. They can also be adapted to each of the various types of adversary query restriction. To save space, we do not present all adaptations, but rather assume the reader will infer them.

4.1.5 Block cipher security

Our main results (Section 4.4) provide explicit reductions from an adversary's advantage against GLEVIAN and VIGORNIAN modes (in a variety of security games) when instantiated with an efficient block cipher $\mathcal{E}$, to an adversary's advantage against $\mathcal{E}$. These results are necessarily computational, so that we may correctly account for the security we inherit from our use of a KDF to provide nonce-dependent keys. We achieve this without recourse to the ideal cipher model (see Section 2 for a brief discussion of the ideal cipher model).

If A is an adversary with run time[16] bounded by t, then A's advantage in distinguishing between a block cipher $\mathcal{E}$ and an RP π is given by

$$\overset{\Delta}{\underset{t}{A}}\begin{pmatrix}\pm\mathcal{E}\\ \pm\pi\end{pmatrix}.$$

In particular, if we wish to secure GLEVIAN and VIGORNIAN modes against adversaries with run time bounded by t (and some fixed query restrictions) then we will see (Section 4.4) that the advantage of A is bounded by a function of their query restrictions, plus a computational $\pm$PRP security expression written as

$$\overset{\Delta}{\underset{t'}{\mathcal{R}(A)}}\begin{pmatrix}\pm\mathcal{E}\\ \pm\pi\end{pmatrix} \tag{3}$$

where $\mathcal{R}$ is the reduction implicit in the proofs in this document and t' is some run time depending on t and on the query restrictions placed on A; see Section 4.4 for details on calculating t' in terms

[16] Recall that we elect not to specify a unit for run time.

of t. Note that we do not specify a query limit; we implicitly grant the adversary 2^n queries so that they may exhaust the block cipher.

We expect GLEVIAN and VIGORNIAN to be instantiated with well-studied secure block ciphers with appropriate key sizes. We also expect that for most practical applications the computational constraints of the $\pm$PRP advantage term (3) are small enough relative to the key size of the block cipher that a very palatable assumption on the block cipher $\mathcal{E}$ renders an adversary's advantage against the entire GLEVIAN or VIGORNIAN system sufficiently small.

Bernstein and Lange [BL13] demonstrated that terms such as (3) can be surprisingly large when quantified over *all* adversaries whose computational resources are bounded in various metrics. We argue that these adversaries do not reduce our confidence in the security of GLEVIAN or VIGORNIAN instances. Our security proof is constructive (that is, given an adversary attacking GLEVIAN or VIGORNIAN instantiated with $\mathcal{E}$ in some security model we can efficiently construct an adversary achieving high $\pm$PRP advantage against $\mathcal{E}$). As the PRP adversaries described in [BL13] are efficient to run but are *not* efficient to find, our argument is essentially the same as the "human ignorance" argument of Rogaway in [Rog06] and so we only describe it briefly here:

1. For a well-studied block cipher that is believed to be secure, we do not believe humans can construct an efficient algorithm with high distinguishing advantage.
2. If a human constructs an efficient attack against GLEVIAN or VIGORNIAN when instantiated with such a block cipher then they can also efficiently construct a distinguisher against the block cipher, contradicting (1).
3. Thus no attack against GLEVIAN or VIGORNIAN can be constructed by humans.

4.1.6 Key to diagrams

We shall make extensive use of diagrams throughout the proofs to describe the components of GLEVIAN and VIGORNIAN. The diagrams will use the following symbols throughout the document:

- $\|$ denotes concatenation
- $(\cdot)$ denotes multiplication in the Galois field $GF(2^n)$
- $\otimes$ also denotes multiplication in the Galois field $GF(2^n)$ where a dot is not suitably clear (i.e. because two lines are coming together)
- $\boxplus$ denotes integer addition modulo 2^n
- $Trunc$ denotes truncation, where the amount to be truncated depends on the context
- Nodes labelled with *split* split the input into multiple output values in ways which are hopefully clear from context.

The diagrams will additionally use the following colour and shape conventions throughout the document:

- Functions shall have rounded corners; values shall have sharp corners
- Thick arrows shall denote 'wide' quantities which may be arbitrarily many bits, thin arrows denote 'narrow' quantities that are exactly a single block in length
- Message or input values shall be blue
- Ciphertext or output values shall be yellow
- Associated data/header/tweak values shall be green
- Key values shall be pink
- Nonce/IV values shall be orange
- Block cipher functions shall be red
- Hash functions shall be cyan.

4.2 Subcomponents of GLEVIAN and VIGORNIAN

Our security proofs for GLEVIAN and VIGORNIAN are inherently modular. We idealise each of the components in turn, measuring or bounding the advantage cost in each case, to bound the overall security of GLEVIAN and VIGORNIAN.

Most of our results are information theoretic. It is only when we introduce the key derivation function in Section 4.2.8 that we switch to computational results.

Throughout this section, π will stand for an RP on n-bit blocks. When some component Π uses an RP as a suboracle we might write $\Pi[\pi]$, or when the dependence on π is clear from context, we just write Π.

4.2.1 CTR

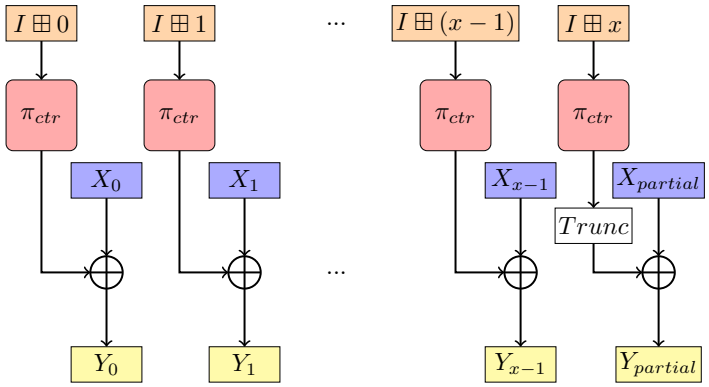


Figure 3: CounTeR mode, denoted CTR$[\pi_{ctr}]$. Diagram symbols and colour conventions are described in Section 4.1.6. Encrypts an input X to an output Y of the same length, under nonce I. Encryption begins by splitting the input into a sequence of blocks and an optional final partial block, as shown in the diagram. $Trunc$ denotes truncation, and is used to make the length of the final block of keystream match the length of $U_{partial}$. The user must be extremely nonce-respecting, i.e. never send a pair of nonces I_1, I_2, each to either oracle, such that $|I_1 \boxminus I_2| \leq \ell$ where ℓ is the maximum allowable message length in blocks. CTR is an involution, so decryption (not shown in the diagram) is identical to encryption.

CounTeR mode (CTR) is described by the diagram in Figure 3. CTR$[\pi_{ctr}]$ expands an n-bit nonce I into a keystream using underlying RP π_{ctr} which is then used to process an input X into an output Y. If the final block is incomplete, the keystream is truncated as required.

The CPA security of CTR is well known. We give the result in our syntax below.

Lemma 9 ($\pm$TPRF security of CTR)**.** *Let π_{ctr} be an RP, and let F be a length-preserving TRF. Let ℓ be the maximum allowable message length in blocks. Let A be an adversary that is extremely nonce-respecting, in the sense that A never queries any pair of (nonce, message) inputs (I, X) and (I', X') to either oracle where the sequences $I, I+1, \ldots, I+(\ell-1)$ and $I', I'+1, \ldots, I'+(\ell-1)$ overlap. Suppose that A can make at most q queries. Then we have that*

$$\Delta_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{CTR}\,[\pi_{ctr}] \\ \pm F\end{matrix}\right) \leq \frac{\ell^2 q^2}{2^{n+1}}.$$

Proof. Apply Lemma 7 (PRP-PRF switch), then direct reasoning. Based on ideas in [BDJR97, Theorem 13]. See Appendix A.2 for a detailed proof. □

Remark 10. This is the only component for which we do not allow the adversary to repeat nonces.

4.2.2 GLEVIAN-HASH

Galois Fields Let $GF(2^n)$ be the Galois field of 2^n elements. Throughout this section and Section 4.2.3, we fix a primitive polynomial $p \in GF(2)[x]$ of degree n, and we let α be a root of p.[17]

[17]To be precise, we assume a polynomial p has been fixed, but do not specify a concrete choice for p in this paper.

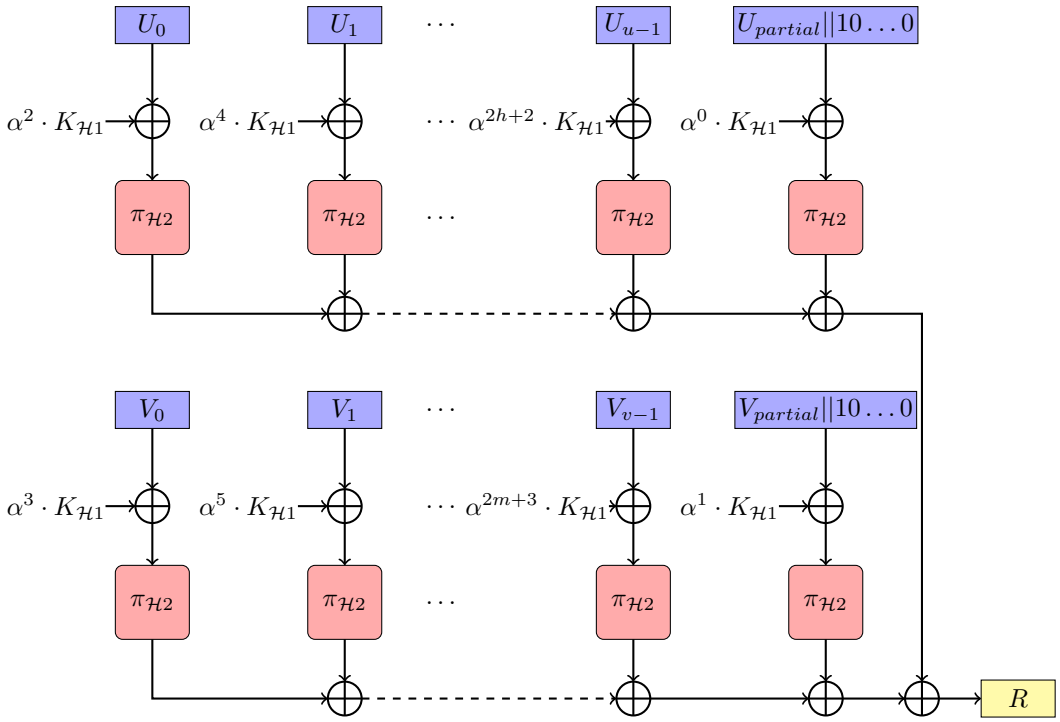


Figure 4: GVN-HASH$_{K_{\mathcal{H}}}[\pi_{\mathcal{H}2}]$, which maps input (U, V) to output R. Diagram symbols and colour conventions are described in Section 4.1.6. First, bitstrings U and V are split into a sequence of whole blocks and an optional partial block, which, if it exists, is processed with 10^* padding. α is chosen to be a fixed primitive element of $GF(2^n)$, and we use $(\cdot)$ to denote multiplication in the Galois Field.

Then α is a primitive element and we can think of the elements of $GF(2^n)$ as polynomials in α of degree less than n. Further, we may interchangeably view elements in the set Σ^n as elements of $GF(2^n)$ by identifying bitstrings $a_0, \ldots, a_{n-1} \in \Sigma^n$ with elements $\sum a_i \alpha^i \in GF(2^n)$. Multiplication of bitstrings will therefore be viewed as multiplication in the Galois field $GF(2^n)$; following standard conventions we shall denote this by any of $\cdot$, $\otimes$ or simply adjacent placement. Multiplication of an n-long bitstring by α exactly corresponds to performing one step of a Linear Feedback Shift Register (LFSR) whose feedback rule corresponds to the coefficients of p.

Definition of GLEVIAN-HASH GLEVIAN-HASH is based on PMAC1 [Rog04a]. GLEVIAN-HASH$_{K_{\mathcal{H}1}}[\pi_{\mathcal{H}2}]$ takes input $(U, V) \in (\Sigma^*)^2$ and computes output R, using an underlying secret permutation $\pi_{\mathcal{H}2}$ and secret key $K_{\mathcal{H}1} \in GF(2^n)$. GLEVIAN-HASH begins by splitting and padding U and M into a sequence of blocks

$$\mathsf{pad}(U, M) = (B_0, B_1, \ldots, B_{b-1}).$$

The B_i are calculated as follows. First, U is split into a sequence of whole blocks $U_0, U_1, \ldots, U_{u-1}$ and an optional partial block U_{partial}. If it exists, U_{partial} is padded with 10^* padding. Then, similarly, V is split into a sequence of whole blocks $V_0, V_1, \ldots, V_{v-1}$ and an optional partial block V_{partial}. If it exists, V_{partial} is padded with 10^* padding. We set $B_0, \ldots, B_{b-1} = U_0, \ldots, U_{u-1}, U_{\text{partial}}, V_0, \ldots, V_{v-1}, V_{\text{partial}}$.

Next, GLEVIAN-HASH assigns an index $\mathsf{index}(B_i)$ to each block $B_i \in \mathsf{pad}(U, V)$. The index function is chosen so that it injectively encodes whether B_i came from U or V, whether B_i came from a whole or partial block, and, when B_i came from a whole block, the block number of that block. The index function is as follows:

$$\mathsf{index}(B_i) = \begin{cases} 0 & \text{if } B_i \text{ came from } U_{\text{partial}} \\ 1 & \text{if } B_i \text{ came from } V_{\text{partial}} \\ 2 + 2j & \text{if } B_i \text{ came from whole block } U_j \\ 3 + 2j & \text{if } B_i \text{ came from whole block } V_j \end{cases}$$

Given the pad and index functions, the output of GLEVIAN-HASH is defined to be

$$\text{GVN-HASH}\,[\pi_{\mathcal{H}2}](U,V) = \bigoplus_{B \in \mathsf{pad}(U,V)} \pi_{\mathcal{H}2}\left(\alpha^{\mathsf{index}(B)} \cdot K \oplus B\right)$$

where addition (denoted $\oplus$), multiplication, and exponentiation are done in $GF(2^n)$.

GLEVIAN-HASH is an ε-AXU$_2$ hash A random variable on the space of functions from arbitrary-length bitstrings to n-long bitstrings with the ε-almost 2-XOR-universal, or ε-AXU$_2$ property [CW79, Rog99], is a family of functions where it is hard for an adversary to construct two messages whose outputs have a known difference, without knowledge of the key or the ability to query the function in advance. The ε-AXU$_2$ property is a slight generalisation of collision resistance and XOR-universal hashing that codifies the security requirement we place on the GLEVIAN-HASH construction.

Definition 11 (ε-AXU$_2$ property)**.** *Let $\mathcal{H}$ be a random variable on the space of functions from bitstrings of length at most ℓ (as defined in Section 4.1.1) to Σ^n. Let ε be a small real number. Then we say that $\mathcal{H}$ has the ε-AXU$_2$ property if for all messages $u \neq v$ and output differences d,*

$$\mathbb{P}\left[\mathcal{H}(u) \oplus \mathcal{H}(v) = d\right] \leq \varepsilon,$$

where the probability is taken over the choice of $\mathcal{H}$. We shall subsequently say that for a hash function $\mathcal{H}$, "$\mathcal{H}$ is an ε-AXU$_2$ for $\epsilon = \ldots$" if it has been sampled in this way.

We will often consider ℓ a variable (rather than a constant as in definition 11). When we do so, we extend definition 11 in the natural way. When doing so, we write $\epsilon = \varepsilon_\ell$ to make clear any dependence of ϵ on ℓ.

Lemma 12 (GLEVIAN-HASH has the ε-AXU$_2$ property)**.** *Let $\pi_{\mathcal{H}2}$ be an RP, and let $K_{\mathcal{H}1}$ be a uniform random variable over $GF(2^n)$. Let $(U,V) \neq (U',V')$ be a pair of inputs, each of total block length at most $\ell \geq 1$ blocks (where lengths are as defined in Section 4.1.1). Let d be an arbitrary output difference. Then*

$$\mathbb{P}[\text{GVN-HASH}_{K_{\mathcal{H}1}}[\pi_{\mathcal{H}2}](U,V) \oplus \text{GVN-HASH}_{K_{\mathcal{H}1}}[\pi_{\mathcal{H}2}](U',V') = d] \leq \frac{5\ell^2}{2^{n+1}}.$$

where the probability is taken over the choice of $\pi_{\mathcal{H}2}$ and $K_{\mathcal{H}1}$.

Proof. Adapted from [Rog04a, Corollary 17]. See Appendix A.3.

4.2.3 VIGORNIAN-HASH

VIGORNIAN-HASH is a polynomial hash taking two inputs U and V and is modelled closely on the well-known GHASH hash function of GCM. Polynomial hashes are described further in [Saa12, dB93, Tay94, BJKS94]. We differ from GHASH in that (for performance reasons applicable to our context) we include the encoded bit-length of the first input U directly after its contents, instead of waiting until the end of the second input V; also, we leave unspecified the block size n and the generating polynomial for the field $GF(2^n)$. In the common case $n = 128$, the field-generating polynomial of GHASH would be a reasonable choice; alternatively, performance improvement might be possible by incorporating the ideas of AES-GCM-SIV's POLYVAL construction.

Definition of VIGORNIAN-HASH VIGORNIAN-HASH is a polynomial hash that computes output R from input $(U,V) \in (\Sigma^*)^2$ and secret key $K_{\mathcal{H}} \in GF(2^n)$. VIGORNIAN-HASH$_{K_{\mathcal{H}}}$ begins by splitting and padding U and V into a sequence of blocks

$$\mathsf{pad}(U,V) = (B_0, B_1, \ldots, B_{b-1}).$$

The B_i are calculated as follows. First, U is split into a sequence of whole n-bit blocks $U_0, U_1, \ldots, U_{u-1}$ and an optional partial final block U_{partial}. If U_{partial} exists it is 0^* padded. We then append a final block $len(U)$ which encodes the bit-length of U. V is similarly split into a sequence of whole blocks

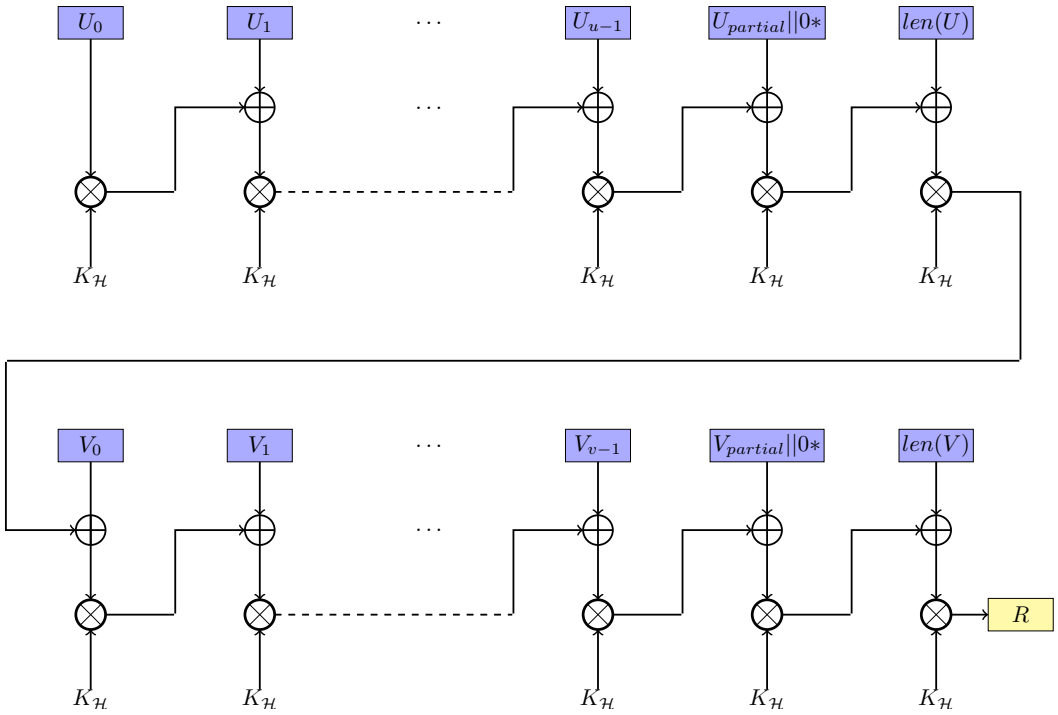


Figure 5: VIGORNIAN-HASH maps input (U, V) to output R under key $K_{\mathcal{H}}$. Diagram symbols and colour conventions are described in Section 4.1.6. First, bitstrings U and V are split into a sequence of blocks. The final partial block of U (resp. V), if it exists, is padded with zeroes, and then an encoding of its bit-length is appended as shown in the diagram.

$V_0, V_1, \ldots, V_{v-1}$ and an optional partial final block V_{partial} with 0^* padding and a terminating block $len(V)$ encoding the bit-length of V. We then define the output of $\mathsf{pad}(U, V)$ by

$$B_{b-1}, B_{b-2}, \ldots, B_0 = U_0, U_1, \ldots, len(U), V_0, V_1, \ldots, len(V).$$

We interpret each B_i as an element of $GF(2^n)$ and set $\mathsf{poly}_{U,V} \in GF(2^n)[x]$ as

$$\mathsf{poly}_{U,V}(x) = \sum_{0 \le i \le b} B_i \cdot x^{i+1}.$$

Finally, $\text{VGN-HASH}_{K_{\mathcal{H}}}(U, V) = \mathsf{poly}_{U,V}(K_{\mathcal{H}})$.

Lemma 13 (VIGORNIAN-HASH has the ε-AXU$_2$ property)**.** *Let $K_{\mathcal{H}}$ be a uniform random variable over $GF(2^n)$. Let $(U, V) \neq (U', V')$ be a pair of inputs, each of total length at most $\ell \geq 1$ blocks (where lengths are as defined in Section 4.1). Let d be an arbitrary output difference. Then*

$$\mathbb{P}[\text{VGN-HASH}_{K_{\mathcal{H}}}(U, V) \oplus \text{VGN-HASH}_{K_{\mathcal{H}}}(U', V') = d] \leq \frac{(\ell + 2)}{2^n},$$

where the probability is taken over the choice of $K_{\mathcal{H}}$.

Proof. See Appendix A.4. □

4.2.4 Wide-tweak narrow-block cipher construction via LRW2

The LRW2 [LRW02] construction is described by the diagram in Figure 6. In the encrypt direction, M is an n-bit input, C is an n-bit output, and the tweak is T which is an arbitrary number of bits wide. As usual, π_{lrw2} is an RP. Let $\mathcal{H}$ be an ε-AXU$_2$, and suppose $\mathcal{H}$ is independent of π_{lrw2}. Given these components, the encryption of message M with tweak T under the LRW2 construction is given by

$$\text{LRW2}\left[\pi_{lrw2}, \mathcal{H}\right](T, M) = \mathcal{H}(T) \oplus \pi_{lrw2}\left(\mathcal{H}(T) \oplus M\right).$$

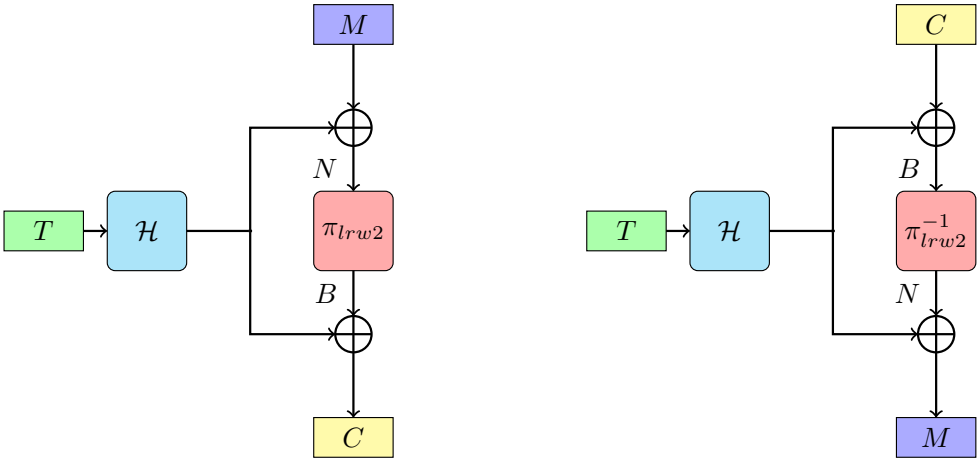


Figure 6: Diagram of the LRW2 $[\pi_{lrw2}, \mathcal{H}]$ construction in the encrypt (left) and decrypt (right) directions. Diagram symbols and colour conventions are described in Section 4.1.6. LRW2$[\pi_{lrw2}, \mathcal{H}]$ encrypts M to C under tweak T as shown in the diagram. N and B label intermediate values as they appear in Figure 7. $\mathcal{H}$ is an ε-AXU$_2$ hash function which is independent of π_{lrw2}.

The decryption of C under tweak T is defined analogously. Later, we will instantiate $\mathcal{H}$ with GLEVIAN-HASH or VIGORNIAN-HASH.

One should have in mind $\mathcal{H}$ being instantiated with one of $\mathcal{H} = \text{GVN-HASH}_{K_{\mathcal{H}1}}[\pi_{\mathcal{H}2}]$ or $\mathcal{H} = \text{VGN-HASH}_{K_{\mathcal{H}}}$, where the randomness is over the choice of key $K_{\mathcal{H}1}$ and block cipher $\pi_{\mathcal{H}2}$ in GLEVIAN-HASH, or just over the choice of key $K_{\mathcal{H}}$ in VIGORNIAN-HASH.

In Figure 7 we present pseudocode that simultaneously defines a lazily sampled implementation of LRW2 and an ideal wide-tweak narrow-block cipher. We present this code here as the analysis of LRW2 will turn out to be critical in achieving our strongest bounds (in Section 4.3.2), and in particular we will require two technical lemmas (Lemmas 14 and 15) that refer specifically to the pseudocode of Figure 7. These results could reasonably be skipped on a first reading and referred back to during the proof of Theorem 27. We then give our generic ±TPRP security result in Lemma 16.

In Lemmas 14, 15 and 16 we consider LRW2 instantiated with a hash $\mathcal{H}$ that has the ε-AXU$_2$ property for some $\varepsilon = \varepsilon_\ell$, where we make the likely dependence of ε_ℓ on ℓ explicit. The proofs of these lemmas are all in Appendix A.5.

Lemma 14 (Bound on probability that adversary sets bad on query i to LRW2)**.** *Let A be an adversary asking queries with tweaks of length at most ℓ blocks, and let bad_i be the event that the code of Figure 7 reaches line 5 or 9 (i.e. the lines that set* bad *to true) on their i^{th} query (beginning the count from 0). Then*

$$\mathbb{P}[\mathsf{bad}_i] \leq 2i\varepsilon_\ell$$

Lemma 15 (Bound on probability that adversary has set bad after q queries to LRW2)**.** *Let A be an adversary asking queries of with tweaks of length at most ℓ blocks. The probability that* bad *is set after q queries to $\pm\mathcal{O}_1$ as defined in Figure 7 is bounded by*

$$\mathbb{P}[\mathsf{bad}] \leq q(q-1)\varepsilon_\ell$$

More generally, in the event that the adversary has access to R copies of $\pm\mathcal{O}_1$ with q queries to each, the probability that bad *is set is bounded by*

$$\mathbb{P}[\mathsf{bad}] \leq Rq(q-1)\varepsilon_\ell$$

Lemma 16 (±TPRP security of LRW2)**.** *Let π_{lrw2} be an RP, and let $\mathcal{H}$ be ε-AXU$_2$ for some $\varepsilon = \varepsilon_\ell$. Then provided π_{lrw2} and $\mathcal{H}$ are independent, the distinguishing advantage between* LRW2 *and a TRP Π with message space Σ^n and tweak space Σ^* against an adversary A making at most q queries where each tweak has length at most ℓ blocks is bounded by*

$$\mathbf{Adv}^{\pm}_{A,q}\begin{pmatrix}\pm\,\text{LRW2}[\pi_{lrw2}, \mathcal{H}] \\ \pm\Pi\end{pmatrix} \leq q(q-1)\varepsilon_\ell.$$

Proof. Inspired by [LRW02, Theorem 2] and [Min07, Theorem 1]. See Appendix A.5. □

Algorithm 2 $\mathcal{O}_1(T, M)$ $\boxed{\mathcal{O}_2(T, M)}$

$C \xleftarrow{\$} \Sigma^n \setminus \mathrm{Im}\,(\Pi(T, \cdot))$
$N \leftarrow M \oplus \mathcal{H}(T)$
$B \leftarrow C \oplus \mathcal{H}(T)$
if $N \in \mathrm{Dom}(\pi_{lrw2})$ **then**
    $\mathsf{bad} \leftarrow \mathsf{true}$
    $\boxed{B \leftarrow \pi_{lrw2}(N)}$
    $\boxed{C \leftarrow \mathcal{H}(T) \oplus B}$
else if $B \in \mathrm{Im}(\pi_{lrw2})$ **then**
    $\mathsf{bad} \leftarrow \mathsf{true}$
    $\boxed{B \xleftarrow{\$} \Sigma^n \setminus \mathrm{Im}(\pi_{lrw2})}$
    $\boxed{C \leftarrow \mathcal{H}(T) \oplus B}$
end if
$\pi_{lrw2}(N) \leftarrow B$
$\Pi(T, M) \leftarrow C$
return C

Algorithm 3 $\mathcal{O}_1^{-1}(T, M)$ $\boxed{\mathcal{O}_2^{-1}(T, M)}$

$M \xleftarrow{\$} \Sigma^n \setminus \mathrm{Dom}\,(\Pi(T, \cdot))$
$N \leftarrow M \oplus \mathcal{H}(T)$
$B \leftarrow C \oplus \mathcal{H}(T)$
if $B \in \mathrm{Im}(\pi_{lrw2})$ **then**
    $\mathsf{bad} \leftarrow \mathsf{true}$
    $\boxed{N \leftarrow \pi_{lrw2}^{-1}(B)}$
    $\boxed{M \leftarrow \mathcal{H}(T) \oplus N}$
else if $N \in \mathrm{Dom}(\pi_{lrw2})$ **then**
    $\mathsf{bad} \leftarrow \mathsf{true}$
    $\boxed{N \xleftarrow{\$} \Sigma^n \setminus \mathrm{Dom}(\pi_{lrw2})}$
    $\boxed{M \leftarrow \mathcal{H}(T) \oplus N}$
end if
$\pi_{lrw2}(N) \leftarrow B$
$\Pi(T, M) \leftarrow C$
return M

Figure 7: Pseudocode jointly defining the LRW2 $[\mathcal{H}, \pi_{lrw2}]$ construction and a TRP Π using lazy sampling. $\mathcal{O}_1$, which does not include the boxed statements, implements a TRP. $\mathcal{O}_2$, which includes the boxed statements, implements LRW2 (see Lemma 39). Given a query of the form (T, M), LRW2 encrypts M to C using tweak T. The shared state of $\mathcal{O}_x$ and $\mathcal{O}_x^{-1}$ is initialised so that bad is set to false and π_{lrw2} and Π are the empty map. For brevity, we do not include caching logic to deal with pointless queries.

4.2.5 Wide-tweak wide-block cipher construction via PIV

The Protected-IV (PIV) construction from [ST13] is described by the diagram in Figure 8. PIV $[\mathcal{B}, \mathcal{C}]$ is a wide-tweak wide-block cipher, built from a wide-tweak, narrow-block TPRP $\mathcal{B}$ and a narrow-nonce based wide-block TPRF $\mathcal{C}$ that is not necessarily nonce misuse resistant. Input U is the (arbitrary length) tweak, and $X = X_L \| X_R$ is a bitstring, where $|X_R| = n$ and X_L has an arbitrary length.

Theorem 1 of [ST13] establishes a bound on the $\pm$TPRP security of the PIV construction[18] In our version we instantiate $\mathcal{C}$ with CTR. As we have defined it, CTR is only secure against *extremely* nonce-respecting adversaries, by which we mean adversaries that never query with either oracle a nonce within ℓ of a nonce that was previously used. We provide an adaptation of the security proof for PIV from [ST13].

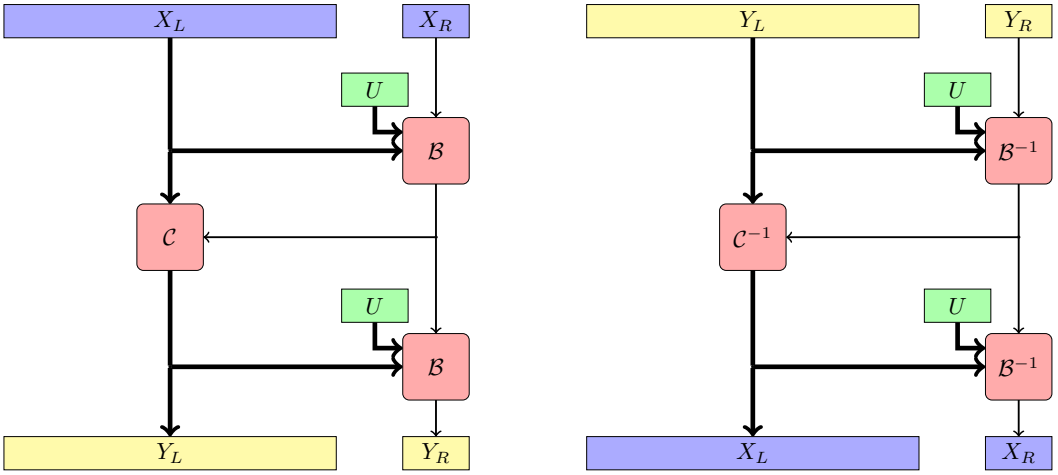
Lemma 17 ($\pm$TPRP security of PIV)**.** *Let A be an adversary that makes at most q queries, where the length after padding of X, Y and U on each query is at most ℓ blocks. Let Ψ be a wide-tweak wide-block TRP, Π be a wide-tweak, narrow-block TRP, and H be a narrow-nonce wide-block TRF. Then we can exhibit a reduction $\mathcal{R}(A)$ and extremely nonce-respecting reductions $\mathcal{S}(A)$ and $\mathcal{T}(A)$, such that $\mathcal{R}(A)$ makes at most $2q$ queries to $\mathcal{B}$ with tweaks of length at most ℓ blocks, $\mathcal{S}(A)$ and $\mathcal{T}(A)$ each make at most q queries of length $\ell - 1$ blocks to $\mathcal{C}$, and*

$$\mathbf{A}^{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm\mathcal{B}, \pm\mathcal{C}] \\ \pm\Psi\end{matrix}\right) \leq \mathbf{R}^{\Delta}_{\mathcal{R}(A),2q}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) + \mathbf{S}^{\Delta}_{\mathcal{S}(A),q,\ell-1}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \mathbf{T}^{\Delta}_{\mathcal{T}(A),q,\ell-1}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \frac{(\ell+9)q^2}{2^{n+1}}.$$

where $\mathcal{R}(A)$ asks queries with tweaks of length at most ℓ blocks.

Proof. An adapted version of [ST13, Theorem 1]. See Appendix A.6. □

[18] Note that there is a minor bug in the proof as originally presented at ASIACRYPT. The event where an attacker collides inputs to the first and second calls to F is not captured in the bound, allowing for something resembling a slide attack. We patch the proof at the same time as adding the modifications described.



(a) PIV Encryption (b) PIV Decryption

Figure 8: The encryption and decryption routines of the PIV construction. Diagram symbols and colour conventions are described in Section 4.1.6. PIV $[\mathcal{B}, \mathcal{C}]$ encrypts input $X = X_L \| X_R$ of minimum length n bits to output $Y = Y_L \| Y_R$ (of the same length) under tweak U. Subcomponent $\mathcal{C}$ should have nonce-respecting $\pm$TPRF security. Subcomponent $\mathcal{B}$ should have $\pm$TPRP security. Decryption (not shown in this diagram) is defined in the obvious way.

Finally, we instantiate subcomponents $\mathcal{B}$ and $\mathcal{C}$ with LRW2 and CTR respectively to form our component WTBC - that is, we set

$$\text{WTBC} = \text{PIV}[\text{LRW2}, \text{CTR}].$$

This construction is pictured in Figure 9. We sometimes make the dependence on the ε-AXU$_2$ component $\mathcal{H}$ and the block cipher π_{lrw2} in LRW2 and on the block cipher π_{ctr} in CTR explicit by writing WTBC $[\mathcal{H}, \pi_{lrw}, \pi_{ctr}]$. We derive a bound on the $\pm$TPRP security of WTBC in the next lemma.

Corollary 18 ($\pm$TPRP security of WTBC)**.** *Let A be an adversary that makes at most q queries, where the length after padding of encryption and decryption queries, and of each tweak, is at most ℓ blocks. Let Ψ be a wide-tweak wide-block TRP, let* LRW2 *be instantiated with an RP π_{lrw2}, let $\mathcal{H}$ be ε-AXU$_2$ for some $\varepsilon = \varepsilon_\ell$ and let* CTR *be instantiated with RP π_{ctr}. Then*

$$\mathbf{Adv}^{\pm}_{A,q,\ell}\begin{pmatrix}\pm\,\text{WTBC}\\ \pm\Psi\end{pmatrix} \leq 4q^2\varepsilon_\ell + \frac{(\ell+2)^2 q^2}{2^n}.$$

Proof. We apply Lemma 17 ($\pm$TPRP security of PIV), and then Lemmas 9 (TPRF security of CTR) and 16 ($\pm$TPRP security of LRW2). See Appendix A.6. □

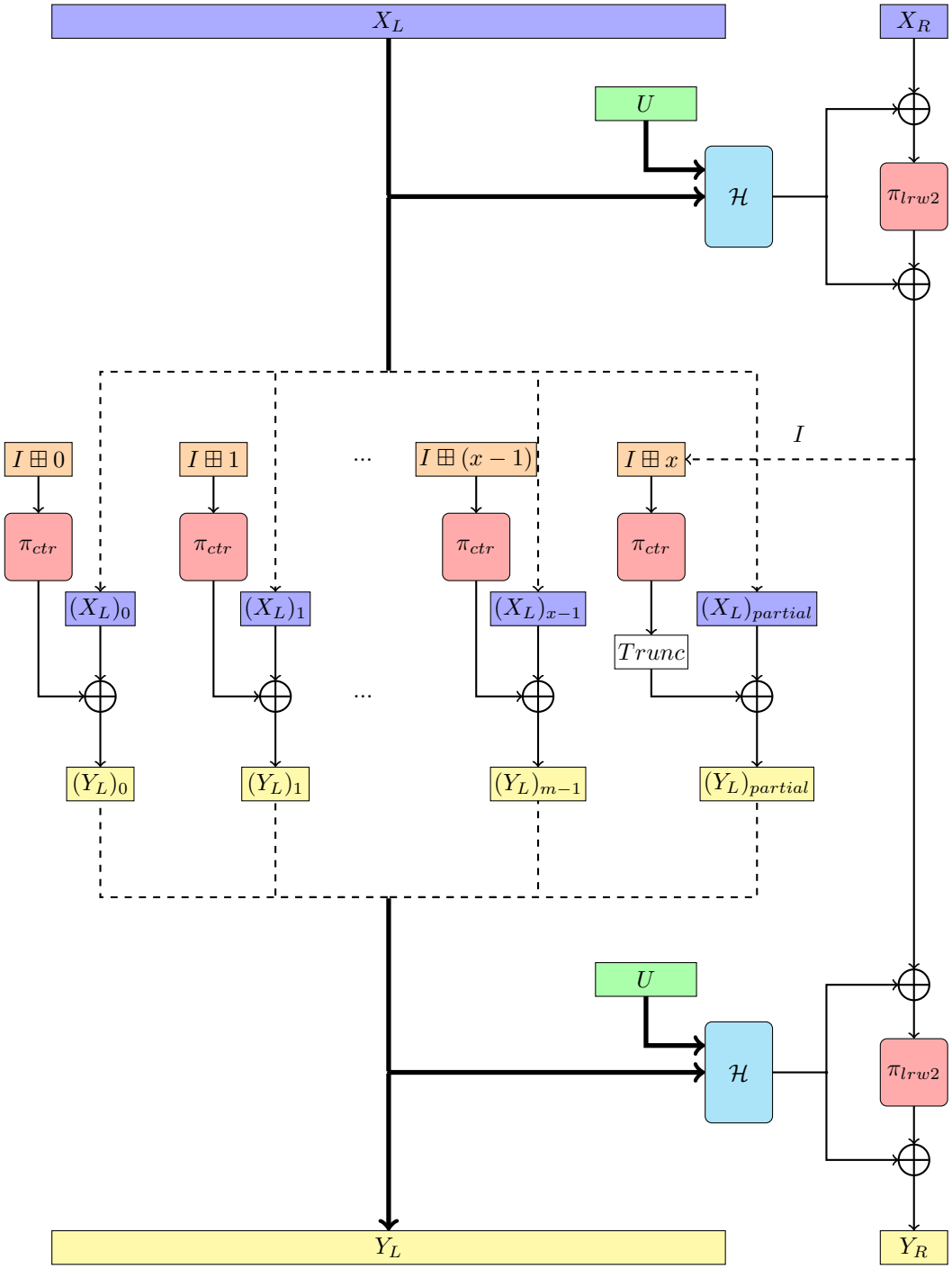
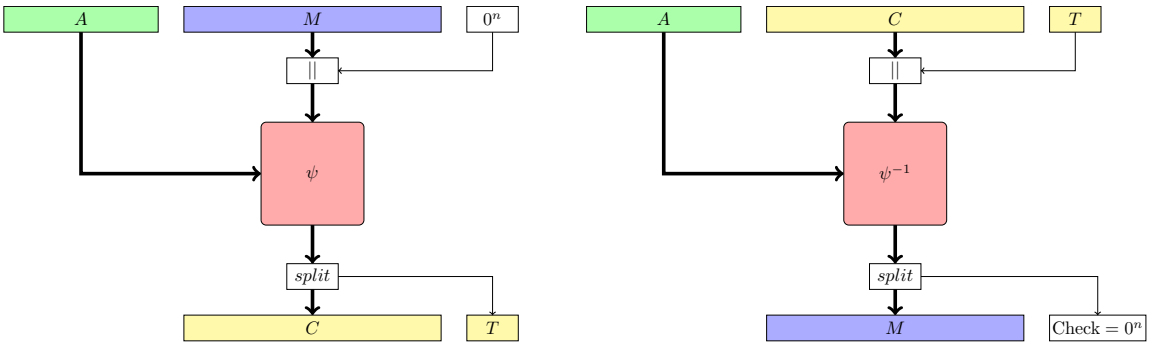


Figure 9: Diagram of the encryption routine for the WTBC$[\mathcal{H}, \pi_{ctr}, \pi_{lrw2}]$ = PIV[CTR$[\pi_{ctr}]$, LRW2$[\pi_{lrw2}, \mathcal{H}]]$ construction. Diagram symbols and colour conventions are described in Section 4.1.6. Decryption (not shown in this diagram)is handled by running encryption in reverse. WTBC stands for Wide Tweakable Block Cipher. WTBC encrypts input $X = X_L \| X_R$ of minimum length n bits to output $Y = Y_L \| Y_R$ (of the same length) under tweak U.



(a) Encryption (b) Decryption

Figure 10: Block diagrams of the encryption/decryption routines for the PTE1 $[\Psi]$ construction. Diagram symbols and colour conventions are described in Section 4.1.6. PTE1 stands for Pad-Then-Encrypt, and implements DAE. Subcomponent Ψ should be a TPRP.

4.2.6 LD-DAE construction via PTE1

Algorithm 4 PTE1 $[\Psi](A, M)$

1: $X \leftarrow M \,\|0^n$
2: $C \leftarrow \Psi(A, X)$
3: **return** C

Algorithm 5 PTE1 $[\Psi]^{-1}(A, C, T)$

1: $Y \leftarrow C \,\|T$
2: $X \leftarrow \Psi^{-1}(A, Y)$
3: $(M \,\|\, \mathsf{check\_zero}) \leftarrow \mathsf{parse}(X)$
4: **if** $\mathsf{check\_zero} = 0^n$ **then**
5: **return** M
6: **else**
7: **return** $\perp$
8: **end if**

Algorithm 6 $\sim$PTE1 $[\Psi](A, C, T)$

1: $Y \leftarrow C \,\|T$
2: $X \leftarrow \Psi^{-1}(A, Y)$
3: $(M \,\|\, \mathsf{check\_zero}) \leftarrow \mathsf{parse}(X)$
4: **if** $\mathsf{check\_zero} = 0^n$ **then**
5: **return** $\top$
6: **else**
7: **return** M
8: **end if**

Figure 11: Pseudocode for the PTE1 $[\Psi]$ construction (encrypt, decrypt, and leakage interfaces). Encrypts message M and associated data A to ciphertext C. Ψ is a wide-tweak wide-block cipher.

The pad-then-encipher (PTE1) construction [RS06] for DAE is pictured in Figures 10a and 10b, with pseudocode appearing in Figure 11. PTE1 encrypts by "padding" (in our case, appending a block of zeroes) a plaintext, and then "enciphering" it. PTE1 differs from the encode-then-encipher construction [BR00] because it contains an associated data input but not a nonce input, thus implementing DAE rather than AEAD. PTE1 encrypts associated data A and message M (each of arbitrary length) to ciphertext C (of the same bit-length as M) and tag T (of bit-length n). 0^n is a bitstring consisting of n zeroes, and *split* denotes the splitting of a bitstring of bit-length $m+n$ into one of bit-length m and one of bit-length n. The "Check= 0^n" redundancy check is intended to provide authentication. If the check fails, PTE1 implementations should return the reject symbol $\perp$ and nothing else.

However, we do still prove that PTE1 is DAE secure even when unverified plaintexts are released. Recall that in the RUP model, the attacker has access to a third "leakage" oracle, which models the fact that real-world decryption implementations may release unverified plaintext. We use the definition of DAE from [RS06, Definition 1], and the notion of extended RUPAE security from [ADL17]. Recall that we justify this choice of RUP model in Section 2.

Lemma 19 (DAE security of PTE1 $[\Psi]$ in the RUP model follows from $\pm$TPRP security of Ψ)**.** *Let A be an adversary making at most q queries each consisting of at most ℓ blocks. Let Ψ be a (not necessarily ideal) wide block cipher, and let Π be an ideal TRP. Let* PTE1 *provide an encrypt, decrypt and leakage interface as defined in Figure 11. Let F be a wide-tweak wide-block stretch-n TRF and*

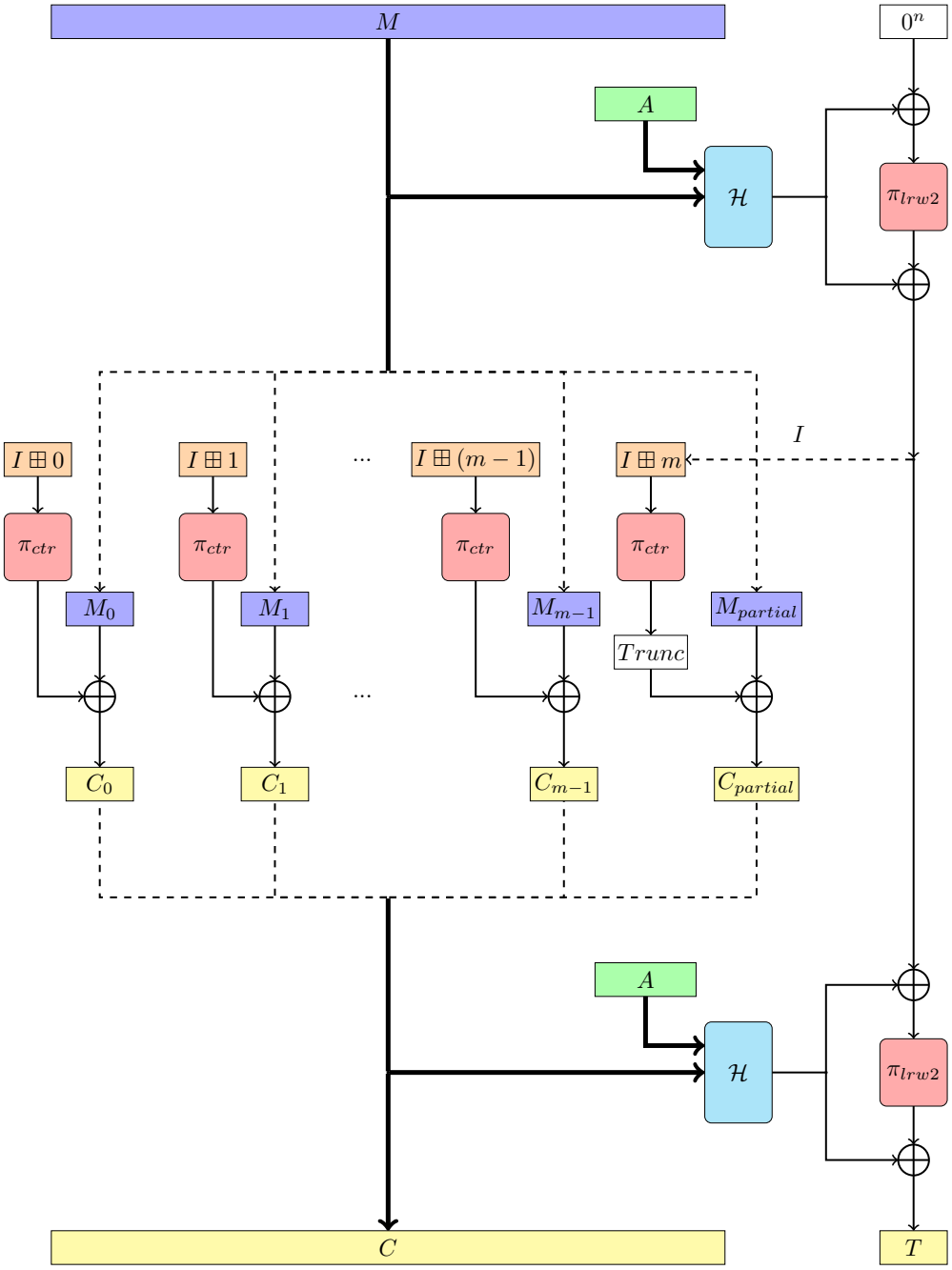


Figure 12: Block diagrams of the encryption routine for the LD-DAE$[\mathcal{H}, \pi_{ctr}, \pi_{lrw2}]$ = PTE1 [WTBC$[\mathcal{H}, \pi_{ctr}, \pi_{lrw2}]$] construction. Diagram symbols and colour conventions are described in Section 4.1.6. In the proof of Theorem 27, notation is abused and the component in this diagram would instead be denoted the LD-DAE[LRW2] construction. LD-DAE stands for Low-Data DAE, and implements authenticated encryption. Our presentation omits a nonce input. LD-DAE encrypts associated data A and message M (each of arbitrary length) to ciphertext C (of the same length as M) and tag T (of length n). 0^n is a bitstring consisting of n zeroes. Decryption implementations should output $\perp$ (if the "Check= 0^n" fails) or M (if the "Check= 0^n" succeeds).

let G be a wide-tweak wide-block shrink-n TRF. Then there is a reduction $\mathcal{R}$ such that

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{A,q,\ell}\left(\begin{matrix}\pm\mathrm{PTE1}\,[\Psi]\\ F,\perp,G\end{matrix}\right) \le \mathbf{Adv}^{\pm\mathrm{prp}}_{\mathcal{R}(A),q,\ell+1}\left(\begin{matrix}\pm\Psi\\ \pm\Pi\end{matrix}\right) + \frac{q^2}{2^n}.$$

Proof. The main ideas and structure of this proof can be found in [RS06, Theorem 11]. We adapt them to our RUP model and present our own proof in Appendix A.7. □

LD-DAE (Low-Data Deterministic Authenticated Encryption, Figure 12) is a DAE component built from the PTE1 construction. That is,

$$\text{LD-DAE} = \mathrm{PTE1}\,[\mathrm{WTBC}\,].$$

When we wish to make clear the dependence of LD-DAE on the ε-AXU$_2$ component $\mathcal{H}$ and the block ciphers π_{lrw2} and π_{ctr} used in Wide-TBC, we write

$$\text{LD-DAE}\,[\mathcal{H},\pi_{lrw2},\pi_{ctr}] = \mathrm{PTE1}[\mathrm{WTBC}\,[\mathcal{H},\pi_{lrw2},\pi_{ctr}]].$$

We conclude that, when instantiated with genuine RPs and either GLEVIAN-HASH or VIGORNIAN-HASH, LD-DAE inherits RUP security from the PTE1 construction. We could have designed an AEAD scheme from LD-DAE by prepending a nonce to the associated data input. We do not directly propose this on account of its birthday security level.

Lemma 20 (DAE security of LD-DAE in the RUP model)**.** *Let π_{lrw2},π_{ctr} be ideal block ciphers and let $\mathcal{H}$ be an ε-AXU$_2$ for some $\varepsilon=\varepsilon_\ell$. Write* LD-DAE *for* LD-DAE $[\mathcal{H},\pi_{lrw2},\pi_{ctr}]$ *and let its leakage interface be as defined in Algorithm 6 (that is, when* ∼LD-DAE *is queried with (A,C,T) it returns all but the last n bits of* WTBC$^{-1}(A,C\|T)$ *whenever* LD-DAE$^{-1}(A,C,T)=\perp$*).*

Let F be a wide-tweak wide-block stretch-n TRF and let G be a wide-tweak wide-block shrink-n TRF. Let A be an adversary making at most q queries each consisting of at most $\ell\ge 1$ blocks.

Then

$$\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{A,q,\ell}\left(\begin{matrix}\pm\text{LD-DAE}\\ F,\perp,G\end{matrix}\right) \le q^2\left(4\varepsilon_\ell + \frac{(\ell+4)^2}{2^n}\right)$$

Proof. Combine Lemma 19 (PTE1 construction) and Corollary 18 (±TPRP security of WTBC). See Appendix A.7. □

4.2.7 KDF via XORP

We convert our birthday-secure DAE scheme LD-DAE to a beyond-birthday secure AEAD scheme by employing the nonce-based key derivation technique of [GLL17]. We instantiate this with the XORP$_w$ construction introduced in [Iwa06] and named in [IMV16, Equation (1)] as a KDF. Our instantiation of XORP$_w$ uses a block cipher π_{xorp} to map an $(n-\lceil\log(w+1)\rceil)$-bit block pseudorandomly onto w n-bit output blocks by

$$\mathrm{XORP}_w[\pi_{xorp}](N) = (\pi_{xorp}(N\|0)\oplus\pi_{xorp}(N\|1),\ldots,\pi_{xorp}(N\|0)\oplus\pi_{xorp}(N\|w))\,.$$

This is illustrated in Figure 13.

Note that this means we have a condition on the width of the block cipher. In order for XORP to generate enough key material, we require enough different inputs to the block cipher to generate at least $w+1$ distinct outputs, i.e. that $n\ge\lceil\log(w+1)\rceil$. For GLEVIAN and VIGORNIAN, we will need $w=1+\left\lceil\frac{3k}{n}\right\rceil$ and $w=1+\left\lceil\frac{2k}{n}\right\rceil$ respectively. In practice, however, this will not be a problem for any conventional block cipher, therefore subsequently we shall assume that n and w always satisfy this inequality.

In order to achieve beyond-birthday security for our extension of LD-DAE we will need XORP$_w$ to achieve beyond-birthday security in the number of queries (although not in w, which should be considered fixed). Both Patarin [Pat10, Theorem 6] and Iwata [Iwa06, Theorem 1] have studied bounds on the security of XORP and found it to achieve beyond-birthday security. We appreciated the commentary on Patarin's proof in [IMV16], but ultimately chose to verify Iwata's less strong bound which still attains adequate beyond-birthday security.

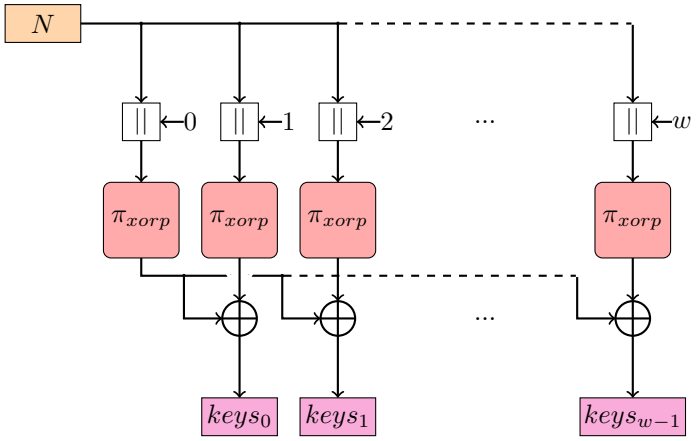


Figure 13: The XORP$[\pi_{xorp}]$ construction takes $(n - \lceil \log(w+1) \rceil)$-bit input N, and produces wn-bit output $keys_0, \ldots, keys_{w-1}$. Diagram symbols and colour conventions are described in Section 4.1.6. ($\|$) denotes concatenation of $(n - \lceil \log(w+1) \rceil)$-bit N with a $\lceil \log(w+1) \rceil$-bit counter; in particular note that the nonce goes first.

Lemma 21 (Security of XORP construction - Iwata 2006)**.** *Let A be an adversary making at most q queries and let F be an RF. Then*

$$\mathop{\Delta}_{A,q}\left(\begin{matrix}\mathrm{XORP}_w[\pi_{xorp}] \\ F\end{matrix}\right) \leq \frac{(w+1)^4 q^3}{2^{2n+1}} + \frac{w(w+1)q}{2^{n+1}}.$$

Proof. See [Iwa06, Theorem 1]. □

4.2.8 Nonce-based key derivation and the HD-AEAD construction

In this section we use the Nonce-based Key Derivation (NKD) paradigm introduced by [GL17] to extend a birthday secure DAE mode to a beyond-birthday secure AEAD mode with the use of a beyond birthday secure KDF. Our security results in this section are necessarily computational as we will use the KDF to provide key material for block cipher(s) in the DAE mode.

We then apply the NKD paradigm to the LD-DAE mode and the XORP KDF. We call the resulting mode HD-AEAD, for High-Data AEAD. GLEVIAN and VIGORNIAN are instantiations of HD-AEAD using different hashes.

Nonce-based key derivation and the HD-AEAD construction NKD is described by the diagram in Figure 16 and fully defined in Figure 14.

Algorithm 7 NKD $[\Pi, \mathcal{K}](N, A, M)$

1: $keys \leftarrow \mathcal{K}(N)$
2: **return** $\Pi_{keys}(A, M)$

Algorithm 8 NKD $[\Pi, \mathcal{K}]^{-1}(N, A, C, T)$

1: $keys \leftarrow \mathcal{K}(N)$
2: **return** $\Pi^{-1}_{keys}(A, C, T)$

Figure 14: Pseudocode for the NKD $[\Pi, \mathcal{K}]$ construction (encrypt and decrypt directions), which is built from a KDF scheme $\mathcal{K}$ and a DAE scheme Π that is initialised with uniformly chosen $keys$. The mode runs in two steps: in the first step, it performs a nonce-based key derivation to produce $keys$. In the second step, the algorithm calls $\pm\Pi$.

Algorithm 9 $\sim\text{NKD}\,[\Pi, \mathcal{K}](N, A, C, T)$

1: $keys \leftarrow \mathcal{K}(N)$
2: **return** $\sim\Pi_{keys}(A, C, T)$

Figure 15: Pseudocode for an example leakage oracle for NKD which forwards the leakage of Π without allowing $\mathcal{K}$ to leak (other leakage oracles could be considered).

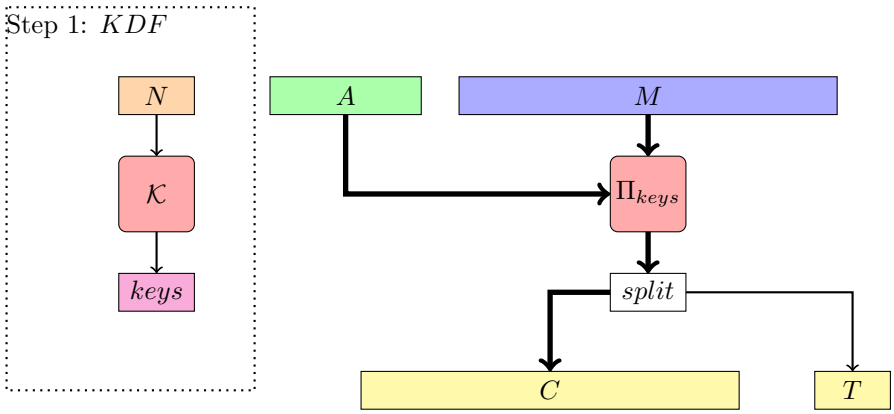


Figure 16: The NKD$[\Pi, \mathcal{K}]$ construction implements nonce-based AEAD. Diagram symbols and colour conventions are described in Section 4.1.6. Named for Nonce-based Key Derivation, in the first step the nonce is processed by the KDF $\mathcal{K}$ to derive per-nonce key material *keys*. In the second step, associated data A and message M are processed by DAE scheme Π under keys *keys*, to produce ciphertext C (of the same length as M) and tag T. Decryption (not shown in the diagram) has the same first step, but the second step is run in reverse.

We show that the security of NKD $[\Pi, \mathcal{K}]$ decomposes into the security of many copies of the underlying DAE scheme Π, and the security of the KDF $\mathcal{K}$. One should keep in mind that Π will often be efficient and, in particular, take a small amount of key material - for example, a series of block cipher keys, as is the case in GLEVIAN and VIGORNIAN. This necessitates consideration of computationally bounded adversaries throughout this section; we make this clear in our lemma statements.

Recall throughout this section that a query restriction of (R, q, f) for an adversary means that adversary may make at most Rq encryption queries using at most R unique nonces and no more than q queries on any given nonce; f decryption queries with free choice of nonce; and f leakage queries with free choice of nonce, if a leakage oracle is available. The related but distinct query restriction of $(R \times q, f)$ for an adversary querying a multi-user oracle $(R + f) \times \pm\mathcal{O}$ or $(R + 2f) \times \underset{\sim}{\pm}\mathcal{O}$ means that adversary may make at most q encryption queries to each of R distinct oracles in the multi-user oracle; and at most f decryption and leakage queries, each of which can be to any of the $R + f$ or $R + 2f$ oracles (depending on whether a leakage interface is available).

Lemma 22 (NKD security reduces to security of the KDF and the underlying DAE scheme)**.** *Let Π be a stretch-n DAE scheme with key space Σ^k for some k. Let $\mathcal{K}$ be a (not necessarily ideal) KDF scheme with codomain Σ^k. Let the leakage interface of Π be arbitrary and for any KDF $\mathcal{K}'$ let the leakage interface of $\text{NKD}\,[\Pi, \mathcal{K}']$ queried with (N, A, C, T) return $\sim\Pi_{\mathcal{K}'(N)}(A, C, T)$ as in Algorithm 9.*

Let F, F' be wide-tweak wide-block stretch-n TRFs and let G, G' be wide-tweak wide-block shrink-n TRFs. Let H be an RF with codomain Σ^k. Let A be an adversary with computational cost bounded by t and with query constraints (R, q, f). Let $s_{\Pi,\ell}$ be the maximum computational cost of simulating a call to any of the interfaces of Π with inputs of length at most ℓ blocks and let s_T be the cost of a lookup into a table of size at most $R + 2f$.

Then there are efficiently constructible reductions $\mathcal{R}, \mathcal{S}$ such that

$$\mathbf{Adv}^{\Delta}_{A,R,q,f,\ell,t}\left(\begin{matrix}\underset{\sim}{\pm}\,\text{NKD}\,[\Pi,\mathcal{K}] \\ F, \perp, G\end{matrix}\right) \leq \mathbf{Adv}^{\Delta}_{\mathcal{R}(A),R+2f,t+(Rq+2f)s_{\Pi,\ell}}\left(\begin{matrix}\mathcal{K} \\ H\end{matrix}\right) + \mathbf{Adv}^{\Delta}_{\mathcal{S}(A),(R\times q),f,\ell,t+(Rq+2f)s_T}\left(\begin{matrix}(R+2f)\times\left(\underset{\sim}{\pm}\Pi\right) \\ (R+2f)\times(F', \perp, G')\end{matrix}\right)$$

Proof. See Appendix A.8. □

HD-AEAD We now define the HD-AEAD (High-Data AEAD) component with respect to an efficient block cipher $\mathcal{E}$ and a keyed hash $\mathcal{H}$ (which one should think of as the ε-AXU$_2$ hash in the LRW2 construction), which we denote HD-AEAD$\,[\mathcal{H}, \mathcal{E}]$.

We require that $\mathcal{H}$ uses at most one block cipher subcomponent (for example, GLEVIAN-HASH uses one, while VIGORNIAN-HASH uses none). We will typically leave the block cipher in $\mathcal{H}$ as implicit, on the understanding that HD-AEAD$\,[\mathcal{H}, \mathcal{E}]$ makes $\mathcal{E}$ available to $\mathcal{H}$ if required; however, where relevant we will make the block cipher available to $\mathcal{H}$ explicit by writing $\mathcal{H}[\mathcal{E}]$ or $\mathcal{H}[\pi]$, even in the case that $\mathcal{H}$ does not depend on a block cipher.

HD-AEAD inherits its key space from the key space of $\mathcal{E}$. For such a key K, define

$$\text{HD-AEAD}\,[\mathcal{H}, \mathcal{E}]_K = \text{NKD}\big[\text{LD-DAE}\,[\mathcal{H}, \mathcal{E}], \text{XORP}_w[\mathcal{E}_{(K)}]\big]$$

where we write XORP$_w[\mathcal{E}_{(K)}]$ for XORP instantiated with $\mathcal{E}$ with key K and LD-DAE$\,[\mathcal{H}, \mathcal{E}]$ for LD-DAE that has been instantiated with all RPs $\pi_{(\cdot)}$ (including any RPs in the as-yet uninstantiated hash component) replaced by block ciphers $\mathcal{E}_{\left(K_{(\cdot)}\right)}$ that are keyed by the output of XORP. That is, for an efficient block cipher $\mathcal{E}$ we write

$$\text{LD-DAE}\,[\mathcal{H}, \mathcal{E}] = \text{LD-DAE}[\mathcal{H}_{(K_{\mathcal{H}})}, \mathcal{E}_{(K_{lrw2})}, \mathcal{E}_{(K_{ctr})}]$$

where each of $K_{\mathcal{H}}, K_{lrw2}, K_{ctr}$ are outputs of XORP and we write $\mathcal{E}_{(K)}$ to mean $\mathcal{E}$ with key K. We write $\mathcal{H}_{(K_{\mathcal{H}})}$ to mean $\mathcal{H}$ implicitly instantiated with block cipher $\mathcal{E}$ and key material $K_{\mathcal{H}}$, which contains all the key material required by $\mathcal{H}$. For example, when $\mathcal{H}$ is instantiated with GLEVIAN-HASH then $K_{\mathcal{H}}$ will consist of an n-bit key and a k-bit key, while if $\mathcal{H}$ is instantiated with VIGORNIAN-HASH then $K_{\mathcal{H}}$ will just be an n-bit key.

HD-AEAD is described in full by the diagram in Figure 17. The security of HD-AEAD reduces immediately to the security of LD-DAE by applying existing lemmas about the security of XORP and of NKD.

Lemma 23 (MRAE security of HD-AEAD from DAE security of LD-DAE)**.** *Let $\mathcal{E}$ be an efficient block cipher and let $\pi, \pi_{\mathcal{H}2}, \pi_{lrw2}, \pi_{ctr}$ be an RP. Let the leakage interface of* LD-DAE *be arbitrary and let* $\sim$HD-AEAD$\,[\mathcal{H}, \mathcal{E}]$ *when queried with (N, A, C, T) be as in Algorithm 9, that is, it returns* $\sim$LD-DAE$\,[\mathcal{H}[\mathcal{E}]_{(K_{\mathcal{H}})}, \mathcal{E}_{(K_{lrw2})}, \mathcal{E}_{(K_{ctr})}](A, C, T)$ *where $K_{\mathcal{H}}, K_{lrw2}, K_{ctr}$ are the output of* XORP$_w[\mathcal{E}](N)$.

Let F, F' be wide-tweak wide-block stretch-n TRFs and let G, G' be wide-tweak wide-block shrink-n TRFs.

Let A be an adversary with computational cost bounded by t and with query constraints (R, q, f). Let s_ℓ be the computational cost of simulating a call to $\underset{\sim}{\pm}$HD-AEAD$\,[\mathcal{H}[\mathcal{E}], \mathcal{E}]$ with inputs of length at most ℓ blocks, or the cost of a lookup into a table of size at most $R + 2f$, whichever is greater.

Then there are efficiently constructible reductions $\mathcal{R}, \mathcal{S}_1, \ldots, \mathcal{S}_4$ such that

$$\mathbf{Adv}^{\pm\text{prp}}_{A,R,q,f,\ell}\left(\begin{matrix}\underset{\sim}{\pm}\,\text{HD-AEAD}\,[\mathcal{H}[\mathcal{E}], \mathcal{E}] \\ F, \bot, G\end{matrix}\right) \le (R+2f)\sum_{i=1}^{4} \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{S}_i(A),\, t+3(Rq+2f)s_\ell}\left(\begin{matrix}\pm\mathcal{E} \\ \pm\pi\end{matrix}\right) \tag{4}$$

$$+ \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \tag{5}$$

$$+ \mathbf{Adv}^{\text{mrae}}_{\mathcal{R}(A),(R\times q),f,\ell}\left(\begin{matrix}(R+2f) \times \big(\underset{\sim}{\pm}\,\text{LD-DAE}\,[\mathcal{H}[\pi_{\mathcal{H}2}], \pi_{lrw2}, \pi_{ctr}]\big) \\ (R+2f) \times (F', \bot, G')\end{matrix}\right)$$

Note that in the computational advantage term here we do not give query constraints and implicitly allow the adversaries 2^n queries; meanwhile, in the information theoretic term we do not give computational constraints and implicitly allow the adversary unbounded computation.

Proof. Based on the ideas of [GL17], but in the computational model. See Appendix A.8 □

Remark 24. Lemma 23 is the source of the multiple block cipher adversaries in our main theorems. As per Section 3.3.1, we endeavour to render all of our adversaries efficiently constructible so that we can argue that the resulting ±PRP advantage terms are as small as possible, at the expense of more unwieldy theorem statements.

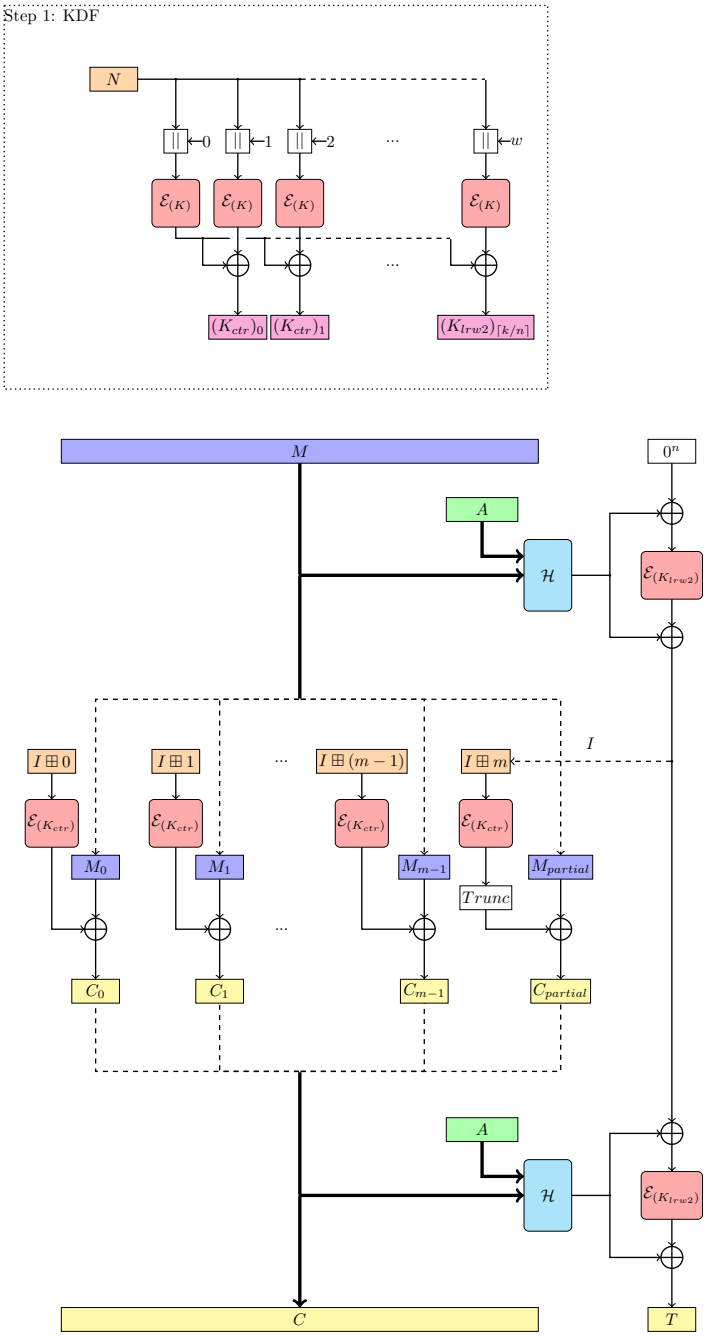


Figure 17: Block diagrams of the encryption routine for the HD-AEAD$_K[\mathcal{E}]$ = NKD[XORP[$\mathcal{E}_K$], LD-DAE] construction. Diagram symbols and colour conventions are described in Section 4.1.6. HD-AEAD stands for High-Data AEAD, and implements authenticated encryption. In the first step the $n - \lceil \log_2(w+1) \rceil$-bit nonce N is processed by the XORP KDF to derive per-nonce key material. In the second step, associated data A and message M are processed by LD-DAE using the derived keys, to produce ciphertext C (of the same length as M) and tag T. Decryption (not shown in the diagram) has the same first step, but the second step is run in reverse. Decryption implementations should output $\perp$ (if the "Check= 0^n" fails) or M (if the "Check= 0^n" succeeds). Note that w is adjusted as needed so that the KDF XORP produces as much output as is needed for various block ciphers and $\mathcal{H}$.

Given parameter choices R, q, f, one can choose a block cipher with a sufficient size key k and make a reasonable security assumption to render terms (4) and (5) negligible. Therefore, to complete our overall advantage bound on HD-AEAD we are left with the problem of establishing a bound on the remaining term:

$$\overset{\Delta}{\mathcal{R}(A),(R\times q),f,\ell}\begin{pmatrix}(R+2f)\times(\underset{\sim}{\pm}\,\text{LD-DAE}\,)\\(R+2f)\times(F,\perp,G)\end{pmatrix} \tag{6}$$

We explore this in the following section, for different choices of $\sim$LD-DAE.

4.3 Building the main theorems - multi-user concentrated forgeries case for LD-DAE

In the previous section, we saw that to complete our overall advantage bound on HD-AEAD we are left with the problem of establishing a bound on the remaining term (6), which we call the *multi-user, concentrated forgeries term.* The name is intended to reflect that in this term, $\mathcal{R}(A)$ is free to concentrate all its decryption queries against a single oracle, or spread them over different oracles – or somewhere in between. We offer two bounds on this term. In the first bound, we allow the adversary access to a leakage interface (i.e. work in the RUP model) and the bound will be cubic in the total number of queries. This will not be a suitable bound for more than a relatively small number of queries, yet it will offer reassurance that the security of HD-AEAD degrades gracefully in the RUP model. In the second bound we do not allow the adversary access to a leakage interface and we obtain a bound linear in $(R + f)$. This demonstrates that our mode retains beyond-birthday security when handling decryption queries correctly.

4.3.1 Loose bound in the RUP model

For the first bound on term (6), we will be in the RUP security model. There is a natural choice for the leakage interface of HD-AEAD: as in Algorithm 9, it inherits its leakage interface from that of LD-DAE and PTE1 as described in Algorithm 6, which can be interpreted as releasing putative or unverified plaintext.

In this model, an adversary that submits all f leakage queries under a single nonce can observe birthday effects in the leaked plaintext. There is therefore no prospect of a bound on (6) in the RUP model that is linear in f. We provide a straightforward but loose security bound where we effectively model the adversary as having f forgery attempts and leakage queries against *each* of the $(R + 2f)$ decryption and leakage oracles, and thus more than f^2 in total. This simple approach allows us to apply Lemma 8 (multi-user to single-user) to find a loose bound.

Lemma 25 (Simple bound on concentrated forgery game in the RUP model)**.** *Let* Π *be a (not necessarily ideal) DAE scheme with arbitrary leakage, let* F *be a wide-tweak wide-block stretch-n TRF and let* G *be a wide-tweak wide-block shrink-n TRF.*

Let A *be an adversary with computational cost bounded by* t *and making at most* $(R\times q), f$ *queries. Let* $s_{\Pi,\ell}$ *be the computational cost of simulating a call to* Π *with inputs of length at most* ℓ *blocks. Then there is an efficiently constructible reduction* $\mathcal{R}$ *such that*

$$\overset{\Delta}{\underset{t}{A,(R\times q),f,\ell}}\begin{pmatrix}(R+2f)\times(\underset{\sim}{\pm}\Pi)\\(R+2f)\times(F,\perp,G)\end{pmatrix} \le (R+2f)\cdot \overset{\Delta}{\underset{t+(Rq+2f)s_{\Pi,\ell}}{\mathcal{R}(A),q+2f,\ell}}\begin{pmatrix}\underset{\sim}{\pm}\Pi\\F,\perp,G\end{pmatrix}$$

Proof. Relax query restrictions to $(R + 2f) \times (q + 2f), \ell$, then apply Lemma 8 (Multi-user to single-user advantage). Note that as A makes at most $Rq + 2f$ queries, $\mathcal{R}$ must simulate (at most) $Rq + 2f$ calls to Π. □

Corollary 26 (Security of HD-AEAD in the RUP model)**.** *Let* $\mathcal{E}$ *be an efficient block cipher, let* $\mathcal{H}$ *be a keyed hash that uses at most one block cipher subcomponent and that has the* ε*-AXU*$_2$ *property for some* $\varepsilon = \varepsilon_\ell$*.*

Let the leakage interface of HD-AEAD $[\mathcal{H}, \mathcal{E}]$ *be defined as in Algorithms 6 and 9 (that is, when* $\sim$HD-AEAD $[\mathcal{H}, \mathcal{E}]$ *is queried with* (N, A, C, T)*, it returns all but the last* n *bits of* WTBC$^{-1}(A, C\|T)$ *whenever* LD-DAE$^{-1}(A, C, T) = \perp$*).*
Let F *be a wide-tweak wide-block stretch-n TRF and let* G *be a wide-tweak wide-block shrink-n TRF.*

Let A be an adversary making at most (R, q, f) queries each consisting of at most ℓ blocks and suppose that $\ell \geq 1$.
Let s_ℓ be the computational cost of simulating a call to $\underset{\sim}{\pm}\,\text{HD-AEAD}\,[\mathcal{H}[\mathcal{E}], \mathcal{E}]$ with inputs of length at most ℓ blocks, or the cost of a lookup into a table of size at most $R + 2f$, whichever is greater.

Then there are efficiently constructible adversaries $\mathcal{R}_1, \ldots, \mathcal{R}_4$ such that

$$\begin{aligned}
\mathbf{Adv}^{\text{MRAE}}_{A,R,q,f,\ell}\left(\begin{matrix}\underset{\sim}{\pm}\,\text{HD-AEAD}\,[\mathcal{H},\mathcal{E}] \\ F, \perp, G\end{matrix}\right) \quad &\leq \quad (R+2f)\sum_{i=1}^{4} \mathbf{Adv}^{\text{PRF}}_{\mathcal{R}_i(A),\, t+3(Rq+2f)s_\ell}\left(\begin{matrix}\pm\mathcal{E} \\ \pm\pi\end{matrix}\right) \\
&+ \quad \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \\
&+ \quad (R+2f)(q+2f)^2 \cdot \left(4\varepsilon_\ell + \frac{(\ell+4)^2}{2^n}\right)
\end{aligned}$$

Proof.

$$\begin{aligned}
\overbrace{\mathbf{Adv}^{\text{MRAE}}_{A,R,q,f,\ell}\left(\begin{matrix}\underset{\sim}{\pm}\,\text{HD-AEAD}\,[\mathcal{H}[\mathcal{E}],\mathcal{E}] \\ F, \perp, G\end{matrix}\right)}^{\substack{\text{Apply Lemma 23} \\ \text{(MRAE security of HD-AEAD} \\ \text{from DAE security of LD-DAE)}}} &\leq (R+2f)\sum_{i=1}^{4} \mathbf{Adv}^{\text{PRF}}_{\mathcal{S}_i(A),\, t+3(Rq+2f)s_\ell}\left(\begin{matrix}\pm\mathcal{E} \\ \pm\pi\end{matrix}\right) \\
&+ \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \\
&+ \mathbf{Adv}^{\text{PRF}}_{\mathcal{R}(A),(R\times q),f,\ell}\left(\begin{matrix}(R+2f)\times\left(\underset{\sim}{\pm}\,\text{LD-DAE}\,[\mathcal{H}[\pi],\pi_{lrw2},\pi_{ctr}]\right) \\ (R+2f)\times(F',\perp,G')\end{matrix}\right)
\end{aligned} \tag{7}$$

where $\pi, \pi_{lrw2}, \pi_{ctr}$ are TRPs and F', G' are TRFs arising from the application of Lemma 23. Apply Lemma 25 (Simple bound on concentrated forgery game in the RUP model) to term (7):

$$\begin{aligned}
(7) &\leq \overbrace{(R+2f)\cdot \mathbf{Adv}^{\text{PRF}}_{\mathcal{R}(A),q+2f,\ell}\left(\begin{matrix}\underset{\sim}{\pm}\,\text{LD-DAE}\,[\mathcal{H}[\pi],\pi_{lrw2},\pi_{ctr}] \\ F', \perp, G'\end{matrix}\right)}^{\substack{\text{Apply Lemma 20} \\ \text{(DAE security of LD-DAE in the RUP model)} \\ \text{with } q+2f \text{ queries}}} \\
&\leq (R+2f)(q+2f)^2\left(4\varepsilon_\ell + \frac{(\ell+4)^2}{2^n}\right)
\end{aligned}$$

This suffices to prove the result. □

Note that for either of GLEVIAN and VIGORNIAN, ε_ℓ never exceeds $\frac{5(\ell+1)^2}{2^n}$ and so this bound gives us reassurance that HD-AEAD retains some security in the RUP model. However, this bound will not be adequate for all use cases. In the next section, we drop the RUP model and our wasteful f^2 modelling to show that the MRAE security of HD-AEAD is linear in $R + f$.

4.3.2 Beyond-birthday bound in the non-RUP model

In the previous section we established a loose bound on the multi-user, concentrated forgeries term (6). This bound was loose because we made the simplifying assumption that the adversary may make f forgery attempts to each of its decrypt oracles. This simplifying assumption arose because we could not rule out that its forgery attempts could be targeted towards one or more somehow "weak" users using information gleaned from encryption queries. However, in the absence of RUP we do not believe that such adaptively forging adversaries are significantly more powerful than ones who do not exercise their adaptive power. Such an adversary would be learning something of the state of an oracle from its encryption queries, which is essentially prohibited by CPA security. Instead, we expect that forgery attempts without RUP will contribute at most linearly to the adversary's advantage (for a justification of why, see e.g. [RS06, Proposition 8]). In this section, we will develop this belief into a concrete argument which will conclude with Theorem 27. Crucially, there is no R^2, Rf, or f^2 term in our bound. We expect that in many practical applications R and f will be quite large, while q and ℓ will be comparatively small, and so Theorem 27 results in a powerful (beyond-birthday) bound.

Regrettably, this proof is less modular than other proofs so far in this paper on account of the tighter reasoning. We hope that anyone looking to adapt this style of proof to their own designs will be able to produce a similar result by following similar reasoning.

Theorem 27 (Multi-user DAE security of LD-DAE)**.** *Let π_{lrw2}, π_{ctr} be ideal block ciphers and let $\mathcal{H}$ be an ε-AXU_2 for some $\varepsilon = \varepsilon_\ell$. Write* LD-DAE *for* LD-DAE$[\mathcal{H}, \pi_{lrw2}, \pi_{ctr}]$*. Let F be a wide-tweak wide-block stretch-n TRF. Let A be an adversary with query constraints $(R \times f), q, f$, where each query is of length at most $\ell \geq 1$ blocks. Then*

$$\mathbf{Adv}^{\mathrm{dae}}_{A,(R\times q),f,\ell}\begin{pmatrix}(R+f)\times(\pm\,\text{LD-DAE})\\(R+f)\times(F,\bot)\end{pmatrix} \leq \frac{1}{2^n}R(\ell+5)^2(q+1)^2 + 12(Rq+f)(q+1)\varepsilon_\ell.$$

Theorem 27 allows us to give a bound for the MRAE security of HD-AEAD. Before proving this theorem, we shall explore the consequences with a corollary and then give some supporting lemmas.

Corollary 28 (MRAE security of HD-AEAD in the non-RUP model)**.** *Let $\mathcal{E}$ be an efficient block cipher and suppose that HD-AEAD does not leak quantities such as unverified plaintext. Let F be a wide-tweak wide-block stretch-n TRF.*

Let A be an adversary with query constraints (R, q, f) and let $\ell, q \geq 1$. Let s_ℓ be the computational cost of simulating a call to $\pm\,\text{HD-AEAD}\,[\mathcal{H}[\mathcal{E}], \mathcal{E}]$ with inputs of length at most ℓ blocks, or the cost of a lookup into a table of size at most $R + 2f$, whichever is greater. Then

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{mrae}}_{A,R,q,f,\ell}\begin{pmatrix}\pm\,\text{HD-AEAD}_K\\F,\bot\end{pmatrix}\\ &\leq (R+2f)\sum_{i=1}^{4}\mathbf{Adv}^{\mathcal{R}_i(A)}_{t+3(Rq+2f)s_\ell}\begin{pmatrix}\pm\mathcal{E}\\\pm\pi\end{pmatrix} + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}}\\ &\quad + \frac{1}{2^n}R(\ell+5)^2(q+1)^2 + 12(Rq+f)(q+1)\varepsilon_\ell.\end{aligned}$$

Proof. Combine Theorem 27 and Lemma 23 (MRAE security of HD-AEAD from DAE security of LD-DAE). □

Second, we shall need some preliminary lemmas about the probability of forgery.

Lemma 29 (Decryption queries at the end)**.** *Let A be an adversary playing G_{dist} between two multi-user worlds of the form $(R + f) \times (\pm\mathcal{O})$ and $(R + f) \times (+\mathcal{P}, \bot)$. Suppose that $\pm\mathcal{O}$ can be eagerly sampled, i.e. the output of each query is independent of previous query inputs. Then there exists an adversary $\mathcal{R}(A)$ which has the same query constraints as A, which is about as efficient to run and has advantage at least as large as A, and which makes all its encryption queries before any decryption queries.*

Proof. See Appendix A.9. □

Note that $\mathcal{R}(A)$ will not necessarily be efficiently constructible from A. Therefore, we are careful to use $\mathcal{R}$ only when reasoning about information-theoretic adversaries and games. In particular, we will never use this lemma as part of a chain of reductions that produces an efficiently constructible block cipher adversary from an efficiently constructible AEAD adversary.

Lemma 30 (G_{dist} as the probability of forgery)**.** *Let A be an adversary playing G_{dist} between two multi-user worlds of the form $(R + f) \times (\pm\mathcal{O})$ and $(R + f) \times (+\mathcal{O}, \bot)$. Suppose that A makes all its encryption queries before its decryption queries. Then*

$$\mathbf{Adv}^{\mathrm{dist}}_{A}\left(\begin{array}{c}(R+f)\times\left(\begin{array}{cc}+\mathcal{O}, & -\mathcal{O}\end{array}\right)\\(R+f)\times\left(\begin{array}{cc}+\mathcal{O}, & \bot\end{array}\right)\end{array}\right) \leq \mathbb{P}[A^{(R+f)\times(\pm\mathcal{O})}\ \textit{forges}].$$

Proof. See Appendix A.9. □

Finally, we now come to prove this section's main theorem.

Proof of Theorem 27. We first apply Lemma 29 (Decryption queries at the end), creating a reduction $\mathcal{R}(\mathrm{A})$ which makes all of its encryption queries before its decryption queries.

Note that since no leakage interface is present, the adversary accesses not $(R + 2f)$ but only $(R + f)$ oracles.

We apply the triangle inequality, decomposing into auth and priv terms:

$$\mathbf{\Delta}_{\mathcal{R}(A),(R\times q),f,\ell}\begin{pmatrix}(R+f)\times(\pm\,\text{LD-DAE}\,)\\(R+f)\times(F,\perp)\end{pmatrix} \le \mathbf{\Delta}_{\mathcal{R}(A),(R\times q),f,\ell}\begin{pmatrix}(R+f)\times(\pm\,\text{LD-DAE}\,)\\(R+f)\times(+\,\text{LD-DAE}\,,\perp)\end{pmatrix} \tag{8}$$

$$+\,\mathbf{\Delta}_{\mathcal{R}(A),(R\times q),f,\ell}\begin{pmatrix}(R+f)\times(+\,\text{LD-DAE}\,,\perp)\\(R+f)\times(F,\perp)\end{pmatrix} \tag{9}$$

1. Privacy term

Term (9) can be bounded easily, because it does not require us to reason about concentrated decryption queries. Let $\mathcal{S}_1(A)$ be the adversary that forwards $\mathcal{R}(A)$'s encryption queries to its encryption oracle, and answers A's decryption queries with $\perp$. Note that since $\mathcal{S}_1(A)$ makes only encryption queries, it accesses only R oracles rather than $(R + f)$. Then $\mathcal{S}_1(A)$ achieves the same advantage as $\mathcal{R}(A)$ does in term (9):

$$(9) = \mathbf{\Delta}_{\mathcal{S}_1(A),(R\times q),\ell}\begin{pmatrix}R\times(+\,\text{LD-DAE}\,)\\R\times F\end{pmatrix}$$

Apply Lemma 8 (Multi-user to single-user advantage), and let $\mathcal{S}_2(A)$ be the adversary derived from $\mathcal{R}(A)$ which plays the single user game:

$$\le R\cdot\mathbf{\Delta}_{\mathcal{S}_2(A),q,\ell}\begin{pmatrix}+\,\text{LD-DAE}\\F\end{pmatrix}$$

Apply Lemma 20 (DAE security of LD-DAE in the RUP model):

$$\le R\left(4q^2\varepsilon_\ell + \frac{(\ell+4)^2q^2}{2^n}\right). \tag{10}$$

2. Authenticity term

We now bound term (8),

$$\mathbf{\Delta}_{\mathcal{R}(A),(R\times q),f,\ell}\begin{pmatrix}(R+f)\times(\pm\,\text{LD-DAE}\,)\\(R+f)\times(+\,\text{LD-DAE}\,,\perp)\end{pmatrix}$$

By Lemma 30, distinguishing the oracles in term (8) will amount to submitting a successful decryption query. One can see from the definition of LD-DAE as a pad-then-encipher scheme that the success of a decryption attempt is closely entwined with the wide-tweak narrow-block cipher component, which in LD-DAE is instantiated with LRW2$[\pi_{lrw2},\mathcal{H}]$. It will become necessary to consider an idealised version of LD-DAE where this entire LRW2 component is replaced by an ideal wide-tweak narrow-block cipher Π. We abuse notation in this section of the proof to write

$$\text{LD-DAE}\,[\text{LRW2}\,] = \text{PTE1}\,[\text{PIV}[\text{LRW2}\,,\text{CTR}\,]]$$
$$\text{LD-DAE}\,[\Pi] = \text{PTE1}\,[\text{PIV}[\Pi,\text{CTR}\,]]\,.$$

Here we have left implicit the dependence of LRW2 and CTR on their subcomponents $\pi_{lrw2},\mathcal{H}$ and π_{ctr} (respectively). In this section of the proof we also consider changes to the encrypt and decrypt interfaces that occur independently; to make this clear, we will make explicit the dependence of the encrypt and decrypt interface of LD-DAE$\,[\cdot]$ on the encrypt and decrypt interface (respectively) of its subcomponent. For example, for LD-DAE [LRW2] we write

$$\pm\,\text{LD-DAE}\,[\text{LRW2}] = (+\,\text{LD-DAE}\,[+\,\text{LRW2}\,],-\,\text{LD-DAE}\,[-\,\text{LRW2}])$$

Finally, we recall the lazily sampled simultaneous implementation of LRW2 and an ideal wide-tweak narrow-block cipher Π in Figure 7. We will refer to this implementation as we complete the proof.

In this notation we can now write term (8) as:

$$\mathcal{R}(A),\overset{\Delta}{(R\times q)},f,\ell \begin{pmatrix} (R+f)\times(\pm\,\text{LD-DAE}\,) \\ (R+f)\times(+\,\text{LD-DAE}\,,\perp) \end{pmatrix} \tag{11}$$
$$= \mathcal{R}(A),\overset{\Delta}{(R\times q)},f,\ell \begin{pmatrix} (R+f)\times\big(\ +\text{LD-DAE}\,[+\,\text{LRW2}\,], & -\,\text{LD-DAE}\,[-\,\text{LRW2}\,]\ \big) \\ (R+f)\times\big(\ +\text{LD-DAE}\,[+\,\text{LRW2}\,], & \perp\ \big) \end{pmatrix}$$

Note further that we can define a lazily sampled oracle $(+\Pi, -\,\text{LRW2}\,)$ that is implemented according to the pseudocode in Figure 7 that *excludes* the boxed statements when answering encrypt queries but *includes* them when answering decrypt queries. Note that there is no corresponding eager sampling implementation of $(+\Pi, -\,\text{LRW2}\,)$. We are careful to make sure this does not affect the proof. Applying the triangle inequality yields:

$$\leq \mathcal{R}(A),\overset{\Delta}{(R\times q)},f,\ell \begin{pmatrix} (R+f)\times\big(\ +\text{LD-DAE}\,[+\,\text{LRW2}\,], & -\,\text{LD-DAE}\,[-\,\text{LRW2}\,]\ \big) \\ (R+f)\times\big(\ +\text{LD-DAE}\,[+\Pi], & -\,\text{LD-DAE}\,[-\,\text{LRW2}\,]\ \big) \end{pmatrix} \tag{12}$$
$$+ \mathcal{R}(A),\overset{\Delta}{(R\times q)},f,\ell \begin{pmatrix} (R+f)\times\big(\ +\text{LD-DAE}\,[+\Pi], & -\,\text{LD-DAE}\,[-\,\text{LRW2}\,]\ \big) \\ (R+f)\times\big(\ +\text{LD-DAE}\,[+\Pi], & \perp\ \big) \end{pmatrix}$$
$$+ \mathcal{R}(A),\overset{\Delta}{(R\times q)},f,\ell \begin{pmatrix} (R+f)\times\big(\ +\text{LD-DAE}\,[+\Pi], & \perp\ \big) \\ (R+f)\times\big(\ +\text{LD-DAE}\,[+\,\text{LRW2}\,], & \perp\ \big) \end{pmatrix} \tag{13}$$

Let bad be as defined by the implementation of Π in Figure 7, and let bad_{enc} (resp. bad_{dec}) be the event that bad is set on an encryption (resp. decryption) query. Each of terms (12) and (13) compares oracles which are identical-until-bad_{enc}, and so the term is bounded by $\mathbb{P}(\mathsf{bad}_{enc})$. Noting that each query to LD-DAE results in two queries to LRW2 with tweaks of length at most ℓ blocks, and noting that $\mathcal{R}(A)$ makes all encryption queries before any decryption queries, we can replace suboracles and apply the multi-user case of Lemma 15 (Probability of bad_{enc}) to find that each of terms (12) and (13) is bounded by $R(2q)(2q-1)\varepsilon_\ell$. So our three terms are bounded by:

$$\leq 2R(2q)(2q-1)\varepsilon_\ell \tag{14}$$
$$+ \mathcal{R}(A),\overset{\Delta}{(R\times q)},f,\ell \begin{pmatrix} (R+f)\times\big(\ +\text{LD-DAE}\,[+\Pi], & -\,\text{LD-DAE}\,[-\,\text{LRW2}\,]\ \big) \\ (R+f)\times\big(\ +\text{LD-DAE}\,[+\Pi], & \perp\ \big) \end{pmatrix} \tag{15}$$

2.(a) Idealised encryption oracles

It remains to bound term (15). Recall that $\mathcal{R}(A)$ makes all of its encryption queries before its decryption queries. So, applying Lemma 30 (G_{dist} as the probability of forgery), we can rewrite this as $\mathcal{R}(A)$'s probability of forging:

$$(15) = \mathbb{P}\left[\mathcal{R}(A)^{(R+f)\times(+\,\text{LD-DAE}\,[+\Pi],-\,\text{LD-DAE}\,[-\,\text{LRW2}\,])}\ \text{forges}\right]$$

For neatness we shall omit the exponent in $\mathcal{R}(A)^{(R+f)\times(+\,\text{LD-DAE}\,[+\Pi],-\,\text{LD-DAE}\,[-\,\text{LRW2}\,])}$ until it becomes relevant again.

$$\leq \mathbb{P}\left[\mathcal{R}(A)\ \text{forges} \mid \overline{\mathcal{R}(A)\ \text{sets}\ \mathsf{bad}_{enc}}\right] + \mathbb{P}\left[\mathcal{R}(A)\ \text{sets}\ \mathsf{bad}_{enc}\right]$$

We bound the probability that $\mathcal{R}$(A) sets bad_{enc} with Lemma 15 (Bound on probability that adversary has set bad after q queries to LRW2):

$$\leq \mathbb{P}\left[\mathcal{R}(A)\ \text{forges} \mid \overline{\mathcal{R}(A)\ \text{sets}\ \mathsf{bad}_{enc}}\right] + R(2q)(2q-1)\varepsilon_\ell$$

2.(a)(i) Forgery without bad_{enc}
It remains to bound the first term. Recalling that $\mathcal{R}(A)$ makes f forgery attempts, we apply a union bound:

$$\mathbb{P}\left[\mathcal{R}(A) \text{ forges} \mid \overline{\mathcal{R}(A) \text{ sets } \mathsf{bad}_{enc}}\right] = \mathbb{P}\left[\bigvee_{1\le j\le f} \mathcal{R}(A) \text{ forges on attempt } j \mid \overline{\mathcal{R}(A) \text{ sets } \mathsf{bad}_{enc}}\right]$$
$$\le \sum_{1\le j\le f} \mathbb{P}\left[\mathcal{R}(A) \text{ forges on attempt } j \mid \overline{\mathcal{R}(A) \text{ sets } \mathsf{bad}_{enc}}\right] \quad (16)$$

We now show that the success or failure of a forgery attempt of $\mathcal{R}(A)$ depends only on the contents of the forgery attempt. Recall that $\mathcal{R}(A)$ makes all of its encryption queries before its decryption queries; we say $\mathcal{R}(A)$ runs an *encryption phase* followed by a *decryption phase*. Conditioned on $\overline{\mathsf{bad}_{enc}}$, the internal state of the $-\,\mathrm{LRW2}$ oracle is consistent at the start of the decryption phase, in the sense that it could have been reached by a lazily sampled implementation of $+\,\mathrm{LRW2}$ in the encryption phase. Since $\pm\,\mathrm{LRW2}$ is eagerly sampleable, we can switch to an eager perspective at this stage. Therefore the probability $\mathcal{R}(\mathrm{A})$ successfully forges on its J^{th} query depends only on the contents of the current forgery attempt and on the transcript of the encryption phase. In particular, the order of the forgery attempts does not matter.

Let $\mathcal{T}(A)$ be the adversary that behaves as follows. $\mathcal{T}(A)$ runs $\mathcal{R}(\mathrm{A})$, forwarding encryption queries to its oracle but storing and rejecting all forgery attempts. When $\mathcal{R}(\mathrm{A})$ has terminated, $\mathcal{T}(A)$ uniformly chooses a random $1 \le J \le f$, and submits $\mathcal{R}(\mathrm{A})$'s J^{th} forgery attempt. $\mathcal{T}(A)$ outputs 1 if and only if it forges. Note that by construction, $\mathcal{T}(A)$ makes all of its encryption queries before its decryption query (which we'll use later). Therefore, by the reasoning in the previous paragraph, we can say that

$$\mathbb{P}\left[\mathcal{R}(A) \text{ forges on attempt } j \mid \overline{\mathcal{R}(\mathrm{A}) \text{ sets } \mathsf{bad}_{enc}}\right] = \mathbb{P}\left[\mathcal{T}(A) \text{ forges} \mid \overline{\mathcal{T}(A) \text{ sets } \mathsf{bad}_{enc}}, \mathcal{T}(A) \text{ chooses } J = j\right]$$

Therefore the sum we were evaluating is:

$$\begin{aligned}(16) &= \sum_{1\le j\le f} \mathbb{P}\left[\mathcal{T}(A) \text{ forges} \mid \overline{\mathcal{T}(A) \text{ sets } \mathsf{bad}_{enc}}, \mathcal{T}(A) \text{ chooses } J = j\right] \\ &= f \cdot \sum_{1\le j\le f} \mathbb{P}\left[\mathcal{T}(A) \text{ forges} \mid \overline{\mathcal{T}(A) \text{ sets } \mathsf{bad}_{enc}}, \mathcal{T}(A) \text{ chooses } J = j\right]\mathbb{P}\left[\mathcal{T}(A) \text{ chooses } J = j\right] \\ &= f \cdot \mathbb{P}\left[\mathcal{T}(A) \text{ forges} \mid \overline{\mathcal{T}(A) \text{ sets } \mathsf{bad}_{enc}}\right]\end{aligned}$$

For any events X, Y with $\mathbb{P}[Y] \le \frac{1}{2}$, note that $\mathbb{P}\left[X \mid \overline{Y}\right] \le 2{\cdot}\mathbb{P}[X]$. In our case, $\mathbb{P}[\mathcal{T}(A) \text{ sets } \mathsf{bad}_{enc}] = R(2q)(2q-1)\varepsilon_\ell$ as seen previously, while our theorem bound is vacuous unless this is at most $\frac{1}{2}$. Therefore we can bound our term by

$$\le 2f \cdot \mathbb{P}[\mathcal{T}(A) \text{ forges}]$$

Since $\mathcal{T}(A)$ only makes a single forgery attempt, we only need to consider a single oracle to which no encryption queries were made:

$$\begin{aligned} &= 2f \cdot \left(\mathbb{P}\left[\mathcal{T}(A)^{((R+1)\times \pm\,\text{LD-DAE}\,[+\Pi, -\,\mathrm{LRW2}])} \text{ forges}\right]\right. \\ &\qquad \left. - \mathbb{P}\left[\mathcal{T}(A)^{((R+1)\times \pm\,\text{LD-DAE}\,[\pm\Pi])} \text{ forges}\right]\right) \\ &\quad + 2f \cdot \mathbb{P}\left[\mathcal{T}(A)^{((R+1)\times \pm\,\text{LD-DAE}\,[\pm\Pi])} \text{ forges}\right] \\ &\le 2f \cdot {}_{\mathcal{T}(A),(R\times q),1,l}^{\Delta}\begin{pmatrix}(R+1) & \times & \pm\,\text{LD-DAE}\,[+\Pi, -\,\mathrm{LRW2}] \\ (R+1) & \times & \pm\,\text{LD-DAE}\,[\pm\Pi]\end{pmatrix} \quad (17) \\ &\quad + 2f \cdot \mathbb{P}\left[\mathcal{T}(A)^{((R+1)\times \pm\,\text{LD-DAE}\,[\pm\Pi])} \text{ forges}\right] \quad (18)\end{aligned}$$

First, we bound term (17), by analysing the multi-user world directly. Recall that $\mathcal{T}(A)$ makes all of its encryption queries before it makes its decryption query. Therefore the two worlds are identical-until-bad_{dec}. So we apply Lemma 6 (Fundamental lemma of game-playing) to bound term (17) by $\mathbb{P}[\mathsf{bad}_{dec}]$.

During each query to LD-DAE[$\pm\mathcal{O}$], the $\pm\mathcal{O}$ oracle is called twice. So during the decryption query, the adversary makes queries number $2q$ and $2q+1$ to $-\mathcal{O}$. Lemma 14 (Bound on probability that adversary sets bad on query i to LRW2) shows that $\mathbb{P}[\mathsf{bad}_{dec}] \leq 2i\varepsilon_\ell$ on query i. So $\mathbb{P}[\mathsf{bad}_{dec}] \leq 4(q+1)\varepsilon_\ell$. It follows that term (17) is bounded by

$$8f(q+1)\varepsilon_\ell. \tag{19}$$

2.(a)(i)(A) Forging with forgeries deferred
We bound (18) in the following claim:
Claim:

$$\mathbb{P}\left[\mathcal{T}(A)^{((R+1)\times\pm\,\text{LD-DAE}\,[\pm\Pi])}\ \text{forges}\right] \leq \frac{8(q+1)}{2^n} \tag{20}$$

Proof of Claim: Consider the following game which we call the "TBC game" (Tweakable Block Cipher game). In the encryption phase of the TBC game, the adversary may make $(R \times 2q)$ queries to $(R \times (+\Pi))$. In the decryption phase, the adversary may make 2 decryption queries to any oracle in $((R+1) \times (-\Pi))$ (note, not forgery attempts - the adversary gets to actually see the output of these decryption queries). The adversary wins the TBC game if they either

1. obtain a Π^{-1} output of zero
2. obtain a Π^{-1} output that matches an output of the corresponding Π oracle from the encryption phase.

Pointless queries are not allowed, so note in both cases, the adversary does *not* win the TBC game if the input to the 'winning' decryption query is the output of an encryption query with the same tweak.

Consider the adversary $\mathcal{U}(A)$ which plays the TBC game by running $\mathcal{T}(A)$, then submitting all the queries to $(R+1) \times (\pm\Pi)$ required to simulate the queries to $(R+1) \times (\pm\,\text{LD-DAE}\,[\pm\Pi])$. We will show that $\mathcal{U}(A)$ wins whenever $\mathcal{T}(A)$ does and so

$$\mathbb{P}\left[\mathcal{T}(A)^{((R+1)\times\pm\,\text{LD-DAE}\,[\pm\Pi])}\ \text{forges}\right] \leq \mathbb{P}\left[\mathcal{U}(A)^{((R+1)\times\pm\Pi)}\ \text{wins the TBC game}\right].$$

Suppose that $\mathcal{T}(A)$ produces a forgery A, C, T with derived plaintext M, and suppose that the intermediate value $\Pi^{-1}((A,C),T) = I$. Since $\mathcal{T}(A)$ produces a valid forgery, $\Pi^{-1}((A,M),I) = 0$. Therefore $\mathcal{U}(A)$ wins the TBC game unless it cannot query Π^{-1} with $((H,M),I)$ due to this being a pointless query. This happens if (H,M) and I were the tweak and output of one of $\mathcal{U}(A)$'s encryption queries, so WLOG assume this. Then either:

1. $((A,M),I) = ((A^*,M^*),I^*)$, where $((A^*,M^*),I^*)$ are the associated data, message, and intermediate value for one of $\mathcal{T}$'s encryption queries. If this were the case, the forgery would simply be the decryption of a previously seen LD-DAE encryption and therefore pointless.
2. $((A,M),I) = ((A^*,C^*),T^*)$, where $((A^*,C^*),T^*)$ are the associated data, ciphertext, and tag for one of $\mathcal{T}$'s encryption queries. In this case, the adversary would win the TBC game by the second criterion.

We shall now consider the probability that $\mathcal{U}(A)$ can win the TBC game. Each of $\mathcal{U}(A)$'s two decryption queries has at most $2q+1$ targets which would result in a win of the TBC game: $2q$ from its previous encryption queries and 1 from zero. Since Π is a TRP, we have that for a given tweak, there will be no collisions between distinct decryption inputs. Therefore of the 2^n possible output values, up to $2q+1$ of them may already have been 'reserved' by previous queries (recall that pointless queries are not allowed): $2q$ from the previous encryption queries and 1 from a previous decryption query.

Therefore the probability that the adversary can win the TBC game after two decryption queries is at most $\frac{2(2q+1)}{2^n-(2q+1)}$. Note that the theorem statement is vacuous[19] unless $5^2(q+1)^2/2^n \leq 1$, so we may henceforth assume that this holds. We therefore have that

$$2q+1 < 2(q+1) \leq \frac{2}{5}2^{n/2} \leq \frac{1}{2}2^n$$

$$\mathbb{P}\left[\mathcal{T}(A)^{((R+1)\times\pm\,\text{LD-DAE}\,[\pm\Pi])} \text{ forges}\right] = \frac{2(2q+1)}{2^n-(2q+1)} \leq \frac{4(q+\frac{1}{2})}{\frac{1}{2}2^n} \leq \frac{8(q+1)}{2^n}$$

■

3. Summary
Collecting terms (10), (14), (19), and (20), we have shown that

$$\begin{aligned}\mathbf{Adv}^{\pm\text{prf}}_{A,(R\times q),f,\ell}\begin{pmatrix}(R+f)\times(\pm\,\text{LD-DAE}\,)\\(R+f)\times(F,\perp)\end{pmatrix} \leq & R\left(4q^2\varepsilon_\ell + \frac{(\ell+4)^2q^2}{2^n}\right)\\ & + 2R(2q)(2q-1)\varepsilon_\ell\\ & + 8f(q+1)\varepsilon_\ell\\ & + \frac{8(q+1)}{2^n}\end{aligned}$$

Simplifying:

$$\begin{aligned}\mathbf{Adv}^{\pm\text{prf}}_{A,(R\times q),f,\ell}\begin{pmatrix}(R+f)\times(\pm\,\text{LD-DAE}\,)\\(R+f)\times(F,\perp)\end{pmatrix} \leq & \quad \frac{1}{2^n}\Big(R(\ell+4)^2q^2+8(q+1)\Big) + \quad \Big(R(4q^2+8q^2)+8f(q+1)\Big)\varepsilon_\ell\\ \leq & \quad \frac{1}{2^n}R(q+1)^2\Big((\ell+4)^2+8\Big) \quad + \quad \Big(12Rq^2+8f(q+1)\Big)\varepsilon_\ell\\ \leq & \quad \frac{1}{2^n}R(\ell+5)^2(q+1)^2 \quad + \quad 12\Big(Rq+f\Big)(q+1)\varepsilon_\ell,\end{aligned}$$

which concludes the theorem, with the last step justified because $(\ell+4)^2+8 < (\ell+5)^2$. □

4.4 GLEVIAN and VIGORNIAN - main security theorems

We define our proposed modes GLEVIAN and VIGORNIAN to be HD-AEAD instantiated with $\mathcal{H}$ =GLEVIAN-HASH and $\mathcal{H}$ =VIGORNIAN-HASH. This is illustrated in Figures 1 and 2. When $\mathcal{E}$ is an efficient block cipher we write k for its keysize (so it has a keyspace of size 2^k). Thus, for GLEVIAN, we can calculate the amount of key material output required from XORP to be $w = 1 + \lceil \frac{3k}{n} \rceil$ blocks, and for VIGORNIAN, we require $w = 1 + \lceil \frac{2k}{n} \rceil$ blocks.

Using the lemmas we have built up we can finally prove the bounds on their security, approximations of which we quoted in Section 3. In each case we give a bound on multi-user MRAE security; that is, the adversaries in these theorems have access to K independent oracles (which one might think of as users with independent keys), and to each oracle they can make encryption queries on up to R distinct nonces, making up to q queries on each nonce; and up to f decryption (and, if applicable, leakage) queries under no constraint on nonces.

Throughout this section we use some standard primitives: we write $\mathcal{E}$ for an efficient block cipher on n-bit blocks and with key space Σ^k; and we write F for a wide-tweak wide-block stretch-n TRF, G for a wide-tweak wide-block shrink-n TRF, and $\perp$ for the oracle that always returns $\perp$. We shall also assume that the block cipher is wide enough (i.e. satisfies $n \geq \lceil \log(w+1) \rceil$) as specified in Section 4.2.7. We also assume that $\ell \geq 1$.

[19] If $R = 0$ then WLOG we assume $q = 0$ and so the inequality holds. If $R > 0$, then the inequality is necessary for the advantage bound in the theorem statement to be less than one.

Theorem 31 (Multi-user MRAE security of GLEVIAN in the RUP model)**.** *Let the leakage interface of GLEVIAN be defined as in Algorithms 6 and 9 (that is, when* $\sim$GVN *is queried with* (N, A, C, T)*, it returns all but the last* n *bits of* $\mathrm{WTBC}^{-1}(A, C\|T)$ *whenever* $\mathrm{GVN}^{-1}(N, A, C, T) = \perp$*). Let* A *be an adversary with computational resource bounded by* t *and with query constraints* $K \times (R, q, f)$ *where each query consists of at most* ℓ *blocks. Let* s_ℓ *be the computational cost of simulating* $\pm\!\!\!\sim\mathrm{GVN}[\mathcal{E}]$ *on a query of at most* ℓ *blocks (which we presume to be more expensive than the cost of simulating a call to* F *or* $\perp$*).*

Then there are efficiently constructible adversaries $\mathcal{R}_1(A), \ldots, \mathcal{R}_4(A)$ *such that*

$$
\begin{aligned}
&\mu\textit{MRAE-RUP}(\mathrm{GVN}) \\
&\quad :=_{A,K\times(R,q,f),\ell}^{\Delta,t} \begin{pmatrix} K \times (\pm\!\!\!\sim\mathrm{GVN}[\mathcal{E}]) \\ K \times (F, \perp, G) \end{pmatrix} \\
&\quad \le K \cdot \left((R+2f) \sum_{i=1}^{4} {}^{\Delta,t+(K+3)(Rq+2f)s_\ell}_{\mathcal{R}_i(A)} \begin{pmatrix} \pm\mathcal{E} \\ \pm\pi \end{pmatrix} + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \right. \\
&\qquad \left. + \frac{11}{2^n}(\ell+1)^2(R+2f)(q+2f)^2 \right),
\end{aligned}
$$

where the constant $w = 1 + \left\lceil \frac{3k}{n} \right\rceil$ *is likely to be* $w \le 7$ *in practice.*

Proof. Apply Lemma 8 (Multi-user to single-user advantage), noting that A makes at most $K(Rq+f)$ queries and simulating responses to any of them costs at most s_ℓ. Then apply Corollary 26 (Security of HD-AEAD in the RUP model). Recall that $\mathrm{GVN}[\mathcal{E}] = \text{HD-AEAD}[\text{GVN-HASH}\,[\mathcal{E}], \mathcal{E}]$, where GLEVIAN-HASH is an ε-AXU$_2$ with $\varepsilon = \frac{5\ell^2}{2^{n+1}}$ (by Lemma 12). Thus,

$$
\begin{aligned}
&\mu\text{MRAE-RUP}(\mathrm{GVN}) \\
&\quad :=_{A,K\times(R\,q,f),\ell}^{\Delta,t} \begin{pmatrix} K \times \big(\pm\!\!\!\sim\mathrm{GVN}[\mathcal{E}]\big) \\ K \times (F, \perp, G) \end{pmatrix} \\
&\quad \le K \cdot {}^{\Delta,A,R,q,f,\ell}_{t+K(Rq+2f)s_\ell} \begin{pmatrix} \pm\!\!\!\sim\mathrm{GVN}[\mathcal{E}] \\ F, \perp, G \end{pmatrix} \\
&\quad \le K \cdot \left((R+2f) \sum_{i=1}^{4} {}^{\Delta,t+(K+3)(Rq+2f)s_\ell}_{\mathcal{R}_i(A)} \begin{pmatrix} \pm\mathcal{E} \\ \pm\pi \end{pmatrix} + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \right. \\
&\qquad \left. (R+2f)(q+2f)^2 \cdot \left(4\frac{5\ell^2}{2^{n+1}} + \frac{(\ell+4)^2}{2^n} \right) \right).
\end{aligned}
$$

which suffices to prove the result, noting that $10\ell^2 + (\ell+4)^2 - 11(\ell+1)^2 = -14\ell + 5 < 0$ for $\ell \ge 1$. $\square$

Theorem 32 (Multi-user MRAE security of GLEVIAN in the non-RUP model)**.** *Let* A *be an adversary with computational resource bounded by* t *and with query constraints* $K \times (R, q, f)$ *where each query consists of at most* ℓ n*-bit blocks. Let* s_ℓ *be the computational cost of simulating* $\pm\mathrm{GVN}[\mathcal{E}]$ *on a query of at most* ℓ *blocks (which we presume to be more expensive than the cost of simulating a call to* F *or* $\perp$*).*

Then, there are efficiently constructible adversaries $\mathcal{R}_1(A), \ldots, \mathcal{R}_4(A)$ *such that*

$$
\begin{aligned}
&\mu\textit{MRAE}(\mathrm{GVN}) \\
&\quad :=_{A,K\times(R,q,f),\ell}^{\Delta,t} \begin{pmatrix} K \times (\pm\mathrm{GVN}[\mathcal{E}]) \\ K \times (F, \perp) \end{pmatrix} \\
&\quad \le K \cdot \left((R+2f) \sum_{i=1}^{4} {}^{\Delta,t+(K+3)(Rq+2f)s_\ell}_{\mathcal{R}_i(A)} \begin{pmatrix} \pm\mathcal{E} \\ \pm\pi \end{pmatrix} + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \right. \\
&\qquad \left. + \frac{31}{2^n}(\ell+5)^2(q+1)\big(R(q+1)+f\big) \right).
\end{aligned}
$$

where the constant $w = 1 + \left\lceil \frac{3k}{n} \right\rceil$ *is likely to be* $w \le 7$ *in practice.*

Proof. Apply Lemma 8 (Multi-user to single-user), noting that A makes at most $K(Rq+f)$ queries and simulating responses to any of them costs at most s_ℓ. Then apply Corollary 28 (Security of HD-AEAD in the non-RUP model). Recall that $\mathrm{GVN}[\mathcal{E}] = \text{HD-AEAD}[\text{GVN-HASH}\,[\mathcal{E}], \mathcal{E}]$, where GLEVIAN-HASH is an ε-AXU$_2$ with $\varepsilon = \frac{5\ell^2}{2^{n+1}}$ (by Lemma 12).

$$
\begin{aligned}
&\mu\mathsf{MRAE}(\mathrm{GVN}) \\
&\quad :=_{A,K\times\binom{R}{t}q,f),\ell} \begin{pmatrix} K \times (\pm\,\mathrm{GVN}[\mathcal{E}]) \\ K \times (F, \bot) \end{pmatrix} \\
&\quad \le K \cdot \underset{t+K(Rq+2f)s_\ell}{\overset{\Delta}{_{A,R,q,f,\ell}}} \begin{pmatrix} \pm\,\mathrm{GVN}[\mathcal{E}] \\ F, \bot \end{pmatrix} \\
&\quad \le K \cdot \Bigg((R+2f) \sum_{i=1}^{4} \underset{t+(K+3)(Rq+2f)s_\ell}{\overset{\Delta}{_{\mathcal{R}_i(A)}}} \begin{pmatrix} \pm\mathcal{E} \\ \pm\pi \end{pmatrix} + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \\
&\qquad\qquad + \frac{1}{2^n} R(\ell+5)^2(q+1)^2 + 12(Rq+f)(q+1)\frac{5\ell^2}{2^{n+1}} \Bigg) \\
&\quad \le K \cdot \Bigg((R+2f) \sum_{i=1}^{4} \underset{t+(K+3)(Rq+2f)s_\ell}{\overset{\Delta}{_{\mathcal{R}_i(A)}}} \begin{pmatrix} \pm\mathcal{E} \\ \pm\pi \end{pmatrix} + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \\
&\qquad\qquad + \frac{1}{2^n}(\ell+5)^2(q+1)\Big(R(q+1) + 30(Rq+f) \Big) \Bigg) \\
&\quad \le K \cdot \Bigg((R+2f) \sum_{i=1}^{4} \underset{t+(K+3)(Rq+2f)s_\ell}{\overset{\Delta}{_{\mathcal{R}_i(A)}}} \begin{pmatrix} \pm\mathcal{E} \\ \pm\pi \end{pmatrix} + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \\
&\qquad\qquad + \frac{1}{2^n}(\ell+5)^2(q+1)\cdot 31\Big(R(q+1) + f \Big) \Bigg) .
\end{aligned}
$$

Reordering variables in the final term completes the result. □

Theorem 33 (Multi-user MRAE security of VIGORNIAN in the RUP model)**.** *Let the leakage interface of VIGORNIAN be defined as in Algorithms 6 and 9 (that is, when* $\sim$VGN *is queried with* (N, A, C, T)*, it returns all but the last* n *bits of* $\mathrm{WTBC}^{-1}(A, C\|T)$ *whenever* $\mathrm{VGN}^{-1}(N, A, C, T) = \bot$*). Let* A *be an adversary with computational resource bounded by* t *and with query constraints* $K \times (R, q, f)$ *where each query consists of at most* ℓ *blocks. Let* s_ℓ *be the computational cost of simulating* $\pm\!\!\!\sim\,\mathrm{VGN}[\mathcal{E}]$ *on a query of at most* ℓ *blocks (which we presume to be more expensive than the cost of simulating a call to* F *or* $\bot$*).*

Then, there are efficiently constructible adversaries $\mathcal{R}_1(A), \ldots, \mathcal{R}_4(A)$ *such that*

$$
\begin{aligned}
&\mu\textit{MRAE-RUP}(\mathrm{VGN}) \\
&\quad :=_{A,K\times\binom{R}{t}q,f),\ell} \begin{pmatrix} K \times (\pm\!\!\!\sim\,\mathrm{VGN}[\mathcal{E}]) \\ K \times (F, \bot, G) \end{pmatrix} \\
&\quad \le K \cdot \Bigg((R+2f) \sum_{i=1}^{4} \underset{t+(K+3)(Rq+2f)s_\ell}{\overset{\Delta}{_{\mathcal{R}_i(A)}}} \begin{pmatrix} \pm\mathcal{E} \\ \pm\pi \end{pmatrix} + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \\
&\qquad\qquad + \frac{1}{2^n}(\ell+6)^2(R+2f)(q+2f)^2 \Bigg) .
\end{aligned}
$$

where the constant $w = 1 + \lceil \frac{2k}{n} \rceil$ *is likely to be* $w \le 5$ *in practice.*

Proof. Apply Lemma 8 (Multi-user to single-user), noting that A makes at most $K(Rq+2f)$ queries and simulating responses to any of them costs at most s_ℓ. Then apply Corollary 26 (Security of HD-AEAD in the RUP model). Recall that $\mathrm{VGN}[\mathcal{E}] = \text{HD-AEAD}[\text{VGN-HASH}\,[\mathcal{E}], \mathcal{E}]$, where VIGORNIAN-HASH is an ε-AXU$_2$ with $\varepsilon = \frac{(\ell+2)}{2^n}$ (by Lemma 13), and where for VIGORNIAN we require $w = 1 + \lceil \frac{2k}{n} \rceil$ blocks of output from XORP. Then,

μMRAE-RUP(VGN)

$$
\begin{aligned}
&:=_{A,K\times\binom{R}{t}q,f),\ell}^{\Delta}\begin{pmatrix}K\times(\pm\!\!\!\sim\,\mathrm{VGN}[\mathcal{E}])\\ K\times(F,\bot,G)\end{pmatrix}\\
&\le K\cdot{}_{t+K(Rq+2f)s_\ell}^{A,R,q,f,\ell,\Delta}\begin{pmatrix}\pm\!\!\!\sim\,\mathrm{VGN}[\mathcal{E}]\\ F,\bot,G\end{pmatrix}\\
&\le K\cdot\left((R+2f)\sum_{i=1}^{4}{}_{t+(K+3)(Rq+2f)s_\ell}^{\mathcal{R}_i(A),\Delta}\begin{pmatrix}\pm\mathcal{E}\\ \pm\pi\end{pmatrix}+\frac{(w+1)^4(R+2f)^3}{2^{2n+1}}+\frac{w(w+1)(R+2f)}{2^{n+1}}\right.\\
&\qquad\left.+(R+2f)(q+2f)^2\cdot\left(4\frac{(\ell+2)}{2^n}+\frac{(\ell+4)^2}{2^n}\right)\right),
\end{aligned}
$$

which suffices to prove the result, noting that $4(\ell+2)+(\ell+4)^2<(\ell+6)^2$. □

Theorem 34 (Multi-user MRAE security of VIGORNIAN in the non-RUP model)**.** *Let A be an adversary with computational resource bounded by t and with query constraints $K\times(R,q,f)$ where each query consists of at most ℓ n-bit blocks. Let s_ℓ be the computational cost of simulating $\pm\mathrm{VGN}[\mathcal{E}]$ on a query of at most ℓ blocks (which we presume to be more expensive than the cost of simulating a call to F or $\bot$). Then, there are efficiently constructible adversaries $\mathcal{R}_1(A),\ldots,\mathcal{R}_4(A)$ such that*

$$
\begin{aligned}
&\mu\mathit{MRAE}(\mathrm{VGN})\\
&:=_{A,K\times\binom{R}{t}q,f),\ell}^{\Delta}\begin{pmatrix}K\times(\pm\,\mathrm{VGN})\\ K\times(F,\bot)\end{pmatrix}\\
&\le K\cdot\left((R+2f)\sum_{i=1}^{4}{}_{t+(K+3)(Rq+2f)s_\ell}^{\mathcal{R}_i(A),\Delta}\begin{pmatrix}\pm\mathcal{E}\\ \pm\pi\end{pmatrix}+\frac{(w+1)^4(R+2f)^3}{2^{2n+1}}+\frac{w(w+1)(R+2f)}{2^{n+1}}\right.\\
&\qquad\left.+\frac{1}{2^n}R(\ell+11)^2(q+1)^2+\frac{12}{2^n}(q+1)(\ell+2)f\right),
\end{aligned}
$$

where the constant $w=1+\left\lceil\frac{2k}{n}\right\rceil$ is likely to be $w\le 5$ in practice.

Proof. Apply Lemma 8 (Multi-user to single-user), noting that A makes at most $K(Rq+f)$ queries and simulating responses to any of them costs at most s_ℓ. Then apply Corollary 28 (Security of HD-AEAD in the non-RUP model). Recall that $\mathrm{VGN}[\mathcal{E}]=\mathrm{HD\text{-}AEAD}[\text{VGN-HASH}\,[\mathcal{E}],\mathcal{E}]$, where VIGORNIAN-HASH is an ε-AXU$_2$ with $\varepsilon=\frac{(\ell+2)}{2^n}$ (by Lemma 13). Then,

$$
\begin{aligned}
&\mu\mathsf{MRAE}(\mathrm{VGN})\\
&:=_{A,K\times\binom{R}{t}q,f),\ell}^{\Delta}\begin{pmatrix}K\times(\pm\,\mathrm{VGN}[\mathcal{E}])\\ K\times(F,\bot)\end{pmatrix}\\
&\le K\cdot{}_{t+K(Rq+f)s_\ell}^{A,R,q,f,\ell,\Delta}\begin{pmatrix}\pm\,\mathrm{VGN}[\mathcal{E}]\\ F,\bot\end{pmatrix}\\
&\le K\cdot\left((R+2f)\sum_{i=1}^{4}{}_{t+(K+3)(Rq+2f)s_\ell}^{\mathcal{R}_i(A),\Delta}\begin{pmatrix}\pm\mathcal{E}\\ \pm\pi\end{pmatrix}+\frac{(w+1)^4(R+2f)^3}{2^{2n+1}}+\frac{w(w+1)(R+2f)}{2^{n+1}}\right.\\
&\qquad\left.+\frac{1}{2^n}R(\ell+5)^2(q+1)^2+12(Rq+f)(q+1)\frac{(\ell+2)}{2^n}\right)\\
&\le K\cdot\left((R+2f)\sum_{i=1}^{4}{}_{t+(K+3)(Rq+2f)s_\ell}^{\mathcal{R}_i(A),\Delta}\begin{pmatrix}\pm\mathcal{E}\\ \pm\pi\end{pmatrix}+\frac{(w+1)^4(R+2f)^3}{2^{2n+1}}+\frac{w(w+1)(R+2f)}{2^{n+1}}\right.\\
&\qquad\left.+\frac{1}{2^n}R(\ell+5)(q+1)^2\Big(\ell+5+12\Big)+\frac{12}{2^n}(q+1)(\ell+2)f\right),
\end{aligned}
$$

which suffices to prove the result, noting that $(\ell+5)(\ell+17)<(\ell+11)^2$. □

5 Conclusion

In this paper, we discuss various features offered by block cipher modes of operation, and their relative importance to high-threat deployments of cryptography. We justify that multi-user security, beyond-birthday security, nonce misuse resistance, and robustness to the release of unverified plaintext are some of the most important properties.

We introduce a further requirement that our mode is block cipher agnostic and compatible with AES. We assign a medium priority to proofs in the standard model, security against quantum adversaries, performance, and context commitment and discovery security.

We demonstrate two modes of operation, GLEVIAN (Figure 1) and VIGORNIAN (Figure 2), that fulfill all of our most important requirements. Full security proofs are given in the standard model, for which the main theorems can be found in Sections 3.3 (simplified) and 4.4 (in full).

We encourage further research on whether our efficiency tradeoffs could be improved, or how to satisfy additional security properties, without compromising on our high-priority requirements.

References

[ABL+14] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to Securely Release Unverified Plaintext in Authenticated Encryption. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pages 105–125, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg, Germany. Cited at 2.1.4, 2.1.4, 2.4.1, and 4.1.2.

[ADG+22] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to Abuse and Fix Authenticated Encryption Without Key Commitment. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022: 31st USENIX Security Symposium*, pages 3291–3308, Boston, MA, USA, August 10–12, 2022. USENIX Association. Cited at 2.2.4 and 2.2.4.

[ADL17] Tomer Ashur, Orr Dunkelman, and Atul Luykx. Boosting Authenticated Encryption Robustness with Minimal Modifications. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 3–33, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany. Cited at 2.1.4, 4.1, and 4.2.6.

[AHR+21] Carmine Abate, Philipp G. Haselwarter, Exequiel Rivas, Antoine Van Muylder, Théo Winterhalter, Catalin Hritcu, Kenji Maillard, and Bas Spitters. SSProve: A Foundational Framework for Modular Cryptographic Proofs in Coq. In Ralf Küsters and Dave Naumann, editors, *CSF 2021: IEEE 34th Computer Security Foundations Symposium*, pages 1–15, Virtual Conference, June 21–24, 2021. IEEE Computer Society Press. Cited at 2.1.6.

[AP13] Nadhem J. AlFardan and Kenneth G. Paterson. Lucky Thirteen: Breaking the TLS and DTLS Record Protocols. In *2013 IEEE Symposium on Security and Privacy*, pages 526–540, Berkeley, CA, USA, May 19–22, 2013. IEEE Computer Society Press. Cited at 2.1.4 and 2.3.1.

[AP16] Martin R. Albrecht and Kenneth G. Paterson. Lucky Microseconds: A Timing Attack on Amazon's s2n Implementation of TLS. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 622–643, Vienna, Austria, May 8–12, 2016. Springer, Heidelberg, Germany. Cited at 2.3.1.

[Bar17] Guy Barwell. Subtleties in Authenticated Encryption: Theoretical Models For Practical Issues, 2017. Cited at 4.1.2.

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 259–274, Bruges, Belgium, May 14–18, 2000. Springer, Heidelberg, Germany. Cited at 2.1.1.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom Functions Revisited: The Cascade Construction and Its Concrete Security. In *37th Annual Symposium on Foundations of Computer Science*, pages 514–523, Burlington, Vermont, October 14–16, 1996. IEEE Computer Society Press. Cited at 2.1.1, 8, and 8.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random Oracles in a Quantum World. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany. Cited at 2.2.2.

[BDJR97] Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A Concrete Security Treatment of Symmetric Encryption. In *38th Annual Symposium on Foundations of Computer Science*, pages 394–403, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press. Cited at 2.1.2 and 4.2.1.

[BDPS14] Alexandra Boldyreva, Jean Paul Degabriele, Kenneth G. Paterson, and Martijn Stam. On Symmetric Encryption with Distinguishable Decryption Failures. In Shiho Moriai, editor, *Fast Software Encryption – FSE 2013*, volume 8424 of *Lecture Notes in Computer Science*, pages 367–390, Singapore, March 11–13, 2014. Springer, Heidelberg, Germany. Cited at 2.1.4.

[Ber05] Daniel J Bernstein. The Poly1305-AES message-authentication code. In *International workshop on fast software encryption*, pages 32–49. Springer, 2005. Cited at 2.4.1.

[Ber08] Daniel J Bernstein. ChaCha, a variant of Salsa20. In *Workshop record of SASC*, 2008. Cited at 2.4.1.

[Ber13] Matthias Berg. The foundational cryptography framework. PhD Dissertation, Saarland University. `https://doi.org/10.22028/D29126528`, 2013. Cited at 2.1.6.

[Ber14] Daniel J Bernstein. CAESAR call for submissions, 2014. https://competitions.cr.yp.to/caeser-call.html. Cited at 2.3.3.

[BGHZ11] Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, and Santiago Zanella Béguelin. Computer-Aided Security Proofs for the Working Cryptographer. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 71–90, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Heidelberg, Germany. Cited at 2.1.6.

[BGJZ07] Gilles Barthe, Benjamin Grégoire, R. Janvier, and Santiago Zanella Béguelin. Formal Certification of Code-Based Cryptographic Proofs. Cryptology ePrint Archive, Report 2007/314, 2007. `https://eprint.iacr.org/2007/314`. Cited at 2.1.6.

[BH22] Mihir Bellare and Viet Tung Hoang. Efficient Schemes for Committing Authenticated Encryption. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 845–875, Trondheim, Norway, May 30 – June 3, 2022. Springer, Heidelberg, Germany. Cited at 2.2.4 and 2.2.4.

[BHT18] Priyanka Bose, Viet Tung Hoang, and Stefano Tessaro. Revisiting AES-GCM-SIV: Multi-user Security, Faster Key Derivation, and Better Bounds. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 468–499, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany. Cited at 2.1.1.

[BJKS94] Jürgen Bierbrauer, Thomas Johansson, Gregory Kabatianskii, and Ben Smeets. On Families of Hash Functions via Geometric Codes and Concatenation. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 331–342, Santa Barbara, CA, USA, August 22–26, 1994. Springer, Heidelberg, Germany. Cited at 4.2.3.

[BKN09] Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolic. Distinguisher and Related-Key Attack on the Full AES-256. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Heidelberg, Germany. Cited at 2.2.1.

[BL12] Daniel J. Bernstein and Tanja Lange. Non-uniform cracks in the concrete: the power of free precomputation. Cryptology ePrint Archive, Report 2012/318, 2012. `https://eprint.iacr.org/2012/318`. Cited at 2.

[BL13] Daniel J. Bernstein and Tanja Lange. Non-uniform Cracks in the Concrete: The Power of Free Precomputation. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II*, volume 8270 of *Lecture Notes in Computer Science*, pages 321–340, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany. Cited at 3.3.1, 4.1.5, and 4.1.5.

[BL16] Karthikeyan Bhargavan and Gaëtan Leurent. On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, pages 456–467, Vienna, Austria, October 24–28, 2016. ACM Press. Cited at 2.3.2.

[Bla06] Bruno Blanchet. A Computationally Sound Mechanized Prover for Security Protocols. In *2006 IEEE Symposium on Security and Privacy*, pages 140–154, Berkeley, CA, USA, May 21–24, 2006. IEEE Computer Society Press. Cited at 2.1.6.

[BLS20] David A. Basin, Andreas Lochbihler, and S. Reza Sefidgar. CryptHOL: Game-Based Proofs in Higher-Order Logic. *Journal of Cryptology*, 33(2):494–566, April 2020. Cited at 2.1.6.

[BNT19] Mihir Bellare, Ruth Ng, and Björn Tackmann. Nonces Are Noticed: AEAD Revisited. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 235–265, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany. Cited at 2.3.3.

[BPS15] Guy Barwell, Daniel Page, and Martijn Stam. Rogue Decryption Failures: Reconciling AE Robustness Notions. In Jens Groth, editor, *15th IMA International Conference on Cryptography and Coding*, volume 9496 of *Lecture Notes in Computer Science*, pages 94–111, Oxford, UK, December 15–17, 2015. Springer, Heidelberg, Germany. Cited at 2.1.4 and 2.1.4.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 317–330, Kyoto, Japan, December 3–7, 2000. Springer, Heidelberg, Germany. Cited at 4.2.6.

[BR06] Mihir Bellare and Phillip Rogaway. The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany. Cited at 6, 6, 6, 6, and A.1.

[CDD+19] Donghoon Chang, Nilanjan Datta, Avijit Dutta, Bart Mennink, Mridul Nandi, Somitra Sanadhya, and Ferdinand Sibleyras. Release of Unverified Plaintext: Tight Unified Model and Application to ANYDAE. *IACR Transactions on Symmetric Cryptology*, 2019(4):119–146, 2019. Cited at 2.1.4.

[CHJS13] Debrup Chakraborty, Vicente Hernandez-Jimenez, and Palash Sarkar. Another Look at XCB. Cryptology ePrint Archive, Report 2013/823, 2013. `https://eprint.iacr.org/2013/823`. Cited at 2.1.6.

[CR22] John Chan and Phillip Rogaway. On Committing Authenticated-Encryption. In Vijayalakshmi Atluri, Roberto Di Pietro, Christian Damsgaard Jensen, and Weizhi Meng, editors, *ESORICS 2022: 27th European Symposium on Research in Computer Security, Part II*, volume 13555 of *Lecture Notes in Computer Science*, pages 275–294, Copenhagen, Denmark, September 26–30, 2022. Springer, Heidelberg, Germany. Cited at 2.2.4.

[CW79] Larry Carter and Mark N. Wegman. Universal Classes of Hash Functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979. Cited at 4.2.2.

[dB93] Bert den Boer. A simple and key-economical unconditional authentication scheme. *Journal of Computer Security*, 2:65–71, 1993. URL: `http://cr.yp.to/bib/entries.html#1993/denboer`. Cited at 4.2.3.

[DFNS11] Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition Attacks on Cryptographic Protocols. Cryptology ePrint Archive, Report 2011/421, 2011. `https://eprint.iacr.org/2011/421`. Cited at 2.2.2.

[DGRW18] Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast Message Franking: From Invisible Salamanders to Encryptment. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 155–186, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany. Cited at 2.2.4 and 2.2.4.

[DR11] Thai Duong and Juliano Rizzo. Here come the XOR ninjas, 2011. Cited at 2.1.3.

[Dwo05] Morris Dworkin. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. Technical Report NIST Special Publication (SP) 800-38B, National Institute of Standards and Technology, Gaithersberg,MD, 2005. Cited at 2.1.2.

[FOR17] Pooya Farshim, Claudio Orlandi, and Răzvan Roşie. Security of Symmetric Primitives under Incorrect Usage of Keys. *IACR Transactions on Symmetric Cryptology*, 2017(1):449–473, 2017. Cited at 2.2.4.

[GH03] Yitchak Gertner and Amir Herzberg. Committing Encryption and Publicly-Verifiable SignCryption. Cryptology ePrint Archive, Report 2003/254, 2003. `https://eprint.iacr.org/2003/254`. Cited at 2.2.4.

[GHS16] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic Security and Indistinguishability in the Quantum World. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 60–89, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany. Cited at 2.2.2.

[GIK+] Chun Guo, Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. A security proof for Romulus-T. `https://romulusae.github.io/romulus/docs/Romulus_T_proof.pdf`. Cited at 2.4.1.

[GL17] Shay Gueron and Yehuda Lindell. Better Bounds for Block Cipher Modes of Operation via Nonce-Based Key Derivation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 1019–1036, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. Cited at 4.2.8 and 4.2.8.

[GLL17] Shay Gueron, Adam Langley, and Yehuda Lindell. AES-GCM-SIV: Specification and Analysis. Cryptology ePrint Archive, Report 2017/168, 2017. `https://eprint.iacr.org/2017/168`. Cited at 2.2.1, 2.4.1, 2.4.1, 3.4, and 4.2.7.

[GLR17] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message Franking via Committing Authenticated Encryption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 66–97, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany. Cited at 2.2.4 and 2.2.4.

[HG16] Ekawat Homsirikamol and Kris Gaj. AEZ: Anything-But EaZy in Hardware. In Orr Dunkelman and Somitra Kumar Sanadhya, editors, *Progress in Cryptology - INDOCRYPT 2016: 17th International Conference in Cryptology in India*, volume 10095 of *Lecture Notes in Computer Science*, pages 207–224, Kolkata, India, December 11–14, 2016. Springer, Heidelberg, Germany. Cited at 2.4.1.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust Authenticated-Encryption AEZ and the Problem That It Solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 15–44, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany. Cited at 2.1.4, 2.1.4, 5, and 2.4.1.

[HR03] Shai Halevi and Phillip Rogaway. A Tweakable Enciphering Mode. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 482–499, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Heidelberg, Germany. Cited at 7 and 7.

[HRRV15] Viet Tung Hoang, Reza Reyhanitabar, Phillip Rogaway, and Damian Vizár. Online Authenticated-Encryption and its Nonce-Reuse Misuse-Resistance. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 493–517, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany. Cited at 2.1.3 and 2.1.3.

[IIMP20] Akiko Inoue, Tetsu Iwata, Kazuhiko Minematsu, and Bertram Poettering. Cryptanalysis of OCB2: Attacks on Authenticity and Confidentiality. *Journal of Cryptology*, 33(4):1871–1913, October 2020. Cited at 2.1.6.

[IKMP20] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Duel of the Titans: The Romulus and Remus Families of Lightweight AEAD Algorithms. *IACR Transactions on Symmetric Cryptology*, 2020(1):43–120, 2020. Cited at 2.4.1.

[IMV16] Tetsu Iwata, Bart Mennink, and Damian Vizár. CENC is Optimally Secure. Cryptology ePrint Archive, Report 2016/1087, 2016. `https://eprint.iacr.org/2016/1087`. Cited at 3.2, 4.2.7, and 4.2.7.

[Iwa06] Tetsu Iwata. New Blockcipher Modes of Operation with Beyond the Birthday Bound Security. In Matthew J. B. Robshaw, editor, *Fast Software Encryption – FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 310–327, Graz, Austria, March 15–17, 2006. Springer, Heidelberg, Germany. Cited at 3.2, 3.2, 4.2.7, 4.2.7, and 4.2.7.

[JNP14] Jérémy Jean, Ivica Nikolic, and Thomas Peyrin. Tweaks and Keys for Block Ciphers: The TWEAKEY Framework. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 274–288, Kaoshiung, Taiwan, R.O.C., December 7–11, 2014. Springer, Heidelberg, Germany. Cited at 3.4.

[Jou06] Antoine Joux. Authentication failures in NIST version of GCM. *NIST Request for Comment*, Published 2006. Cited at 2.1.3 and 2.4.1.

[KLLN16] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Quantum Differential and Linear Cryptanalysis. *IACR Transactions on Symmetric Cryptology*, 2016(1):71–94, 2016. `https://tosc.iacr.org/index.php/ToSC/article/view/536`. Cited at 2.2.2.

[KM19] Neal Koblitz and Alfred Menezes. Critical Perspectives on Provable Security: Fifteen Years of "Another Look" Papers. Cryptology ePrint Archive, Report 2019/1336, 2019. `https://eprint.iacr.org/2019/1336`. Cited at 2.1.6.

[KR11] Ted Krovetz and Phillip Rogaway. The Software Performance of Authenticated-Encryption Modes. In Antoine Joux, editor, *Fast Software Encryption – FSE 2011*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327, Lyngby, Denmark, February 13–16, 2011. Springer, Heidelberg, Germany. Cited at 2.4.1.

[LGR21] Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning Oracle Attacks. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021: 30th USENIX Security Symposium*, pages 195–212. USENIX Association, August 11–13, 2021. Cited at 2.2.4.

[LRW02] Moses Liskov, Ronald L. Rivest, and David Wagner. Tweakable Block Ciphers. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 31–46, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Heidelberg, Germany. Cited at 3.2, 4.2.4, 4.2.4, and A.5.

[MDK14] Bodo Moller, Thai Duong, and Krzysztof Kotowicz. This POODLE Bites: Exploiting the SSL 3.0 Fallback. *Google Security Advisory*, 2014. Cited at 2.3.1.

[Min07] Kazuhiko Minematsu. Improved Security Analysis of XEX and LRW Modes. In Eli Biham and Amr M. Youssef, editors, *SAC 2006: 13th Annual International Workshop on Selected Areas in Cryptography*, volume 4356 of *Lecture Notes in Computer Science*, pages 96–113, Montreal, Canada, August 17–18, 2007. Springer, Heidelberg, Germany. Cited at 4.2.4 and A.5.

[Min20] Kazuhiko Minematsu. Fast Decryption: a New Feature of Misuse-Resistant AE. *IACR Transactions on Symmetric Cryptology*, 2020(3):87–118, 2020. Cited at 3.

[MLGR23] Sanketh Menda, Julia Len, Paul Grubbs, and Thomas Ristenpart. Context Discovery and Commitment Attacks - How to Break CCM, EAX, SIV, and More. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV*, volume 14007 of *Lecture Notes in Computer Science*, pages 379–407. Springer, 2023. Cited at 2.2.4 and 2.2.4.

[MV04] David A. McGrew and John Viega. The Security and Performance of the Galois/-Counter Mode (GCM) of Operation. In Anne Canteaut and Kapalee Viswanathan, editors, *Progress in Cryptology - INDOCRYPT 2004: 5th International Conference in Cryptology in India*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355, Chennai, India, December 20–22, 2004. Springer, Heidelberg, Germany. Cited at 2.4.1.

[NIS23] The Third NIST Workshop on Block Cipher Modes of Operation 2023. `https://csrc.nist.gov/events/2023/third-workshop-on-block-cipher-modes-of-operation`, 2023. Cited at 1.

[NL15] Yoav Nir and Adam Langley. ChaCha20 and Poly1305 for IETF Protocols. RFC 7539, May 2015. Cited at 2.4.1.

[Pat10] Jacques Patarin. Introduction to Mirror Theory: Analysis of Systems of Linear Equalities and Linear Non Equalities for Cryptography. Cryptology ePrint Archive, Report 2010/287, 2010. `https://eprint.iacr.org/2010/287`. Cited at 3.2 and 4.2.7.

[PDM+18] Damian Poddebniak, Christian Dresen, Jens Müller, Fabian Ising, Sebastian Schinzel, Simon Friedberger, Juraj Somorovsky, and Jörg Schwenk. Efail: Breaking {S/MIME} and {OpenPGP} Email Encryption using Exfiltration Channels. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 549–566, 2018. Cited at 2.1.4.

[PM15] Adam Petcher and Greg Morrisett. The foundational cryptography framework. In *Principles of Security and Trust: 4th International Conference, POST 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings 4*, pages 53–72. Springer, 2015. Cited at 2.1.6.

[Rog99] Phillip Rogaway. Bucket Hashing and Its Application to Fast Message Authentication. *Journal of Cryptology*, 12(2):91–115, March 1999. Cited at 4.2.2.

[Rog04a] Phillip Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In Pil Joong Lee, editor, *Advances in Cryptology – ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31, Jeju Island, Korea, December 5–9, 2004. Springer, Heidelberg, Germany. Cited at 3.2, 4.2.2, and 4.2.2.

[Rog04b] Phillip Rogaway. Nonce-Based Symmetric Encryption. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 348–359, New Delhi, India, February 5–7, 2004. Springer, Heidelberg, Germany. Cited at 2.1.2.

[Rog06] Phillip Rogaway. Formalizing Human Ignorance. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT 06: 1st International Conference on Cryptology in Vietnam*, volume 4341 of *Lecture Notes in Computer Science*, pages 211–228, Hanoi, Vietnam, September 25–28, 2006. Springer, Heidelberg, Germany. Cited at 4.1.5.

[RPS18] Eyal Ronen, Kenneth G. Paterson, and Adi Shamir. Pseudo Constant Time Implementations of TLS Are Only Pseudo Secure. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 1397–1414, Toronto, ON, Canada, October 15–19, 2018. ACM Press. Cited at 2.3.1.

[RS06] Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany. Cited at 2.1.3, 7, 3.2, 4.1, 4.2.6, 4.2.6, 4.2.6, and 4.3.2.

[Saa12] Markku-Juhani Olavi Saarinen. Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes. In Anne Canteaut, editor, *Fast Software Encryption – FSE 2012*, volume 7549 of *Lecture Notes in Computer Science*, pages 216–225, Washington, DC, USA, March 19–21, 2012. Springer, Heidelberg, Germany. Cited at 4.2.3.

[ST13] Thomas Shrimpton and R. Seth Terashima. A Modular Framework for Building Variable-Input-Length Tweakable Ciphers. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 405–423, Bengalore, India, December 1–5, 2013. Springer, Heidelberg, Germany. Cited at 3.2, 4.2.5, 4.2.5, 4.2.5, and 4.2.5.

[Tay94] Richard Taylor. An Integrity Check Value Algorithm for Stream Ciphers. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 40–48, Santa Barbara, CA, USA, August 22–26, 1994. Springer, Heidelberg, Germany. Cited at 4.2.3.

[VP17] Mathy Vanhoef and Frank Piessens. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 1313–1328, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. Cited at 2.1.3.

[Zha12] Mark Zhandry. How to Construct Quantum Random Functions. In *53rd Annual Symposium on Foundations of Computer Science*, pages 679–687, New Brunswick, NJ, USA, October 20–23, 2012. IEEE Computer Society Press. Cited at 2.2.2.

A Additional proofs

A.1 Proofs of preliminary lemmas

This appendix gives supporting proofs for the results in Section 4.1.4.

Lemma 2 (Adversaries have no need to make pointless queries)**.** *Let A be an adversary, and $\mathcal{O}, \mathcal{P}$ be oracles. Suppose that A makes pointless queries. Then there is a derived adversary $\mathcal{R}(A)$ such that $\mathcal{R}(A)$ makes no more queries than A, does not make pointless queries, and achieves the same advantage as A - i.e.*

$$\overset{\Delta}{A}\begin{pmatrix}\mathcal{O}\\ \mathcal{P}\end{pmatrix} = \overset{\Delta}{\mathcal{R}(A)}\begin{pmatrix}\mathcal{O}\\ \mathcal{P}\end{pmatrix}$$

Proof. Let A be an adversary that achieves some distinguishing advantage between $\mathcal{O}$ and $\mathcal{P}$. We define the derived adversary $\mathcal{R}(A)$ as follows. $\mathcal{R}(A)$ runs A. Each time A makes a query, $\mathcal{R}(A)$ intercepts that query. If it is pointless, $\mathcal{R}(A)$ answers A directly without interacting with the challenge oracle. If it is not pointless, then $\mathcal{R}(A)$ forwards it to the challenge oracle. $\mathcal{R}(A)$ then returns the value which A outputs. $\mathcal{R}(A)$ therefore achieves the same distinguishing advantage as A, without making pointless queries. □

Lemma 3 (Triangle inequality; Δ has metric-like properties)**.** *For any adversary A and oracles $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3$,*

$$\overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_1\\ \mathcal{O}_3\end{pmatrix} \leq \overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_1\\ \mathcal{O}_2\end{pmatrix} + \overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_2\\ \mathcal{O}_3\end{pmatrix}$$

Proof.

$$\begin{aligned}
&\overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_1\\ \mathcal{O}_3\end{pmatrix}\\
&= \left|\mathbb{P}\left[A^{\mathcal{O}_1} \Rightarrow 1\right] - \mathbb{P}\left[A^{\mathcal{O}_3} \Rightarrow 1\right]\right|\\
&= \left|\left(\mathbb{P}\left[A^{\mathcal{O}_1} \Rightarrow 1\right] - \mathbb{P}\left[A^{\mathcal{O}_2} \Rightarrow 1\right]\right) + \left(\mathbb{P}\left[A^{\mathcal{O}_2} \Rightarrow 1\right] - \mathbb{P}\left[A^{\mathcal{O}_3} \Rightarrow 1\right]\right)\right|\\
&\leq \left|\left(\mathbb{P}\left[A^{\mathcal{O}_1} \Rightarrow 1\right] - \mathbb{P}\left[A^{\mathcal{O}_2} \Rightarrow 1\right]\right)\right| + \left|\left(\mathbb{P}\left[A^{\mathcal{O}_2} \Rightarrow 1\right] - \mathbb{P}\left[A^{\mathcal{O}_3} \Rightarrow 1\right]\right)\right|\\
&= \overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_1\\ \mathcal{O}_2\end{pmatrix} + \overset{\Delta}{A,q}\begin{pmatrix}\mathcal{O}_2\\ \mathcal{O}_3\end{pmatrix}.
\end{aligned}$$

□

Lemma 4 (Replacing suboracles)**.** *Let $\mathcal{O}$ be an oracle making use of a suboracle, denoted π. Suppose that a query to $\mathcal{O}$ generates at most r queries to π and suppose $\mathcal{O}$ does not access the internal state of π. Let s be the computational cost of simulating a call to $\mathcal{O}$ given oracle access to π.*

Let π_1, π_2 be two instantiations of π. Then there is a reduction $\mathcal{R}$ such that for any adversary A making at most q queries,

$$\overset{\Delta}{\underset{t}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_1]\\ \mathcal{O}[\pi_2]\end{pmatrix} \leq \overset{\Delta}{\underset{t+sq}{\mathcal{R}(A),rq}}\begin{pmatrix}\pi_1\\ \pi_2\end{pmatrix}.$$

Proof. The reduction $\mathcal{R}$ constructs the adversary $B = \mathcal{R}(A)$ from A as follows. B is initialised with access to an oracle π, and it hopes to distinguish whether $\pi = \pi_1$ or $\pi = \pi_2$. B runs A. Whenever A emits a query, B simulates $\mathcal{O}[\pi]$ using its oracle π as the suboracle of $\mathcal{O}$ and uses the result to answer A. When A emits its guess as to whether it was interacting with $\mathcal{O}[\pi_1]$ or $\mathcal{O}[\pi_2]$, B emits the corresponding guess about whether it was interacting with π_1 or π_2. B has the same advantage as A, since B guesses correctly iff A does. Since A makes at most q queries, and for each query A makes, B makes at most r corresponding queries, B must make at most rq queries. Further, since A runs in at most t units of work and with at most q queries, we have that B runs in at most $t + sq$ units of work as required, since each time A makes an oracle call (costing zero units of work), B must simulate it for s unit of work. □

Corollary 5 (Replacing suboracles)**.** *Let $\mathcal{O}, \pi_1, \pi_2, r, s$ be as above, and let $\mathcal{P}$ be an oracle with the same input and output types as $\mathcal{O}$. Then there is a reduction $\mathcal{R}$ such that for any adversary A making at most q queries*

$$\overset{\Delta}{\underset{t}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_1]\\ \mathcal{P}\end{pmatrix} \leq \overset{\Delta}{\underset{t+sq}{\mathcal{R}(A),rq}}\begin{pmatrix}\pi_1\\ \pi_2\end{pmatrix} + \overset{\Delta}{\underset{t}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_2]\\ \mathcal{P}\end{pmatrix}.$$

Proof. We start with the LHS, and apply Lemma 3 (Triangle inequality):

$$\overset{\Delta}{\underset{t}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_1]\\ \mathcal{P}\end{pmatrix} \leq \overset{\Delta}{\underset{t}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_1]\\ \mathcal{O}[\pi_2]\end{pmatrix} + \overset{\Delta}{\underset{t}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_2]\\ \mathcal{P}\end{pmatrix}$$

Then apply Lemma 4 (Replacing suboracles) to the first term:

$$\leq \overset{\Delta}{\underset{t+sq}{\mathcal{R}(A),rq}}\begin{pmatrix}\pi_1\\ \pi_2\end{pmatrix} + \overset{\Delta}{\underset{t}{A,q}}\begin{pmatrix}\mathcal{O}[\pi_2]\\ \mathcal{P}\end{pmatrix}.$$

□

Lemma 6 (Fundamental lemma of game-playing [BR06])**.** *Let $G^{A,\mathcal{O}}$ and $H^{A,\mathcal{P}}$ be identical-until-*bad*, in the sense of [BR06]. Then*

$$|\mathbb{P}(G^{A,\mathcal{O}} \Rightarrow 1) - \mathbb{P}(H^{A,\mathcal{P}} \Rightarrow 1)| \leq \mathbb{P}[G^{A,\mathcal{O}} \textit{ sets } \mathsf{bad}].$$

In particular, if $G = H = G_{\mathsf{dist}}$, and A is an adversary, then

$$\overset{\Delta}{\underset{A}{}}\begin{pmatrix}\mathcal{O}\\ \mathcal{P}\end{pmatrix} \leq \mathbb{P}[G^{A,\mathcal{O}}_{\mathsf{dist}} \textit{ sets } \mathsf{bad}].$$

Proof. See [BR06, Lemma 2]. □

Algorithm 10 $\mathrm{TRF}(T, X)$ $\boxed{\mathrm{TRP}(T, X)}$

$Y \xleftarrow{\$} \Sigma^n$
if $X \in \mathrm{Dom}_{\mathcal{L}(T,\cdot)}$ **then**
  $Y \xleftarrow{\$} \{Y : (T, X, Y) \in \mathcal{L}\}$
else if $Y \in \mathrm{Ran}_{\mathcal{L}(T,\cdot)}$ **then**
  bad ← true
  $\boxed{Y \xleftarrow{\$} \Sigma^n \setminus \mathrm{Ran}_{\mathcal{L}(T,\cdot)}}$
end if
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(T, X, Y)\}$
return Y

Algorithm 11 $\mathrm{TRF}^{-1}(T, Y)$ $\boxed{\mathrm{TRP}^{-1}(T, Y)}$

$X \xleftarrow{\$} \Sigma^n$
if $Y \in \mathrm{Ran}_{\mathcal{L}(T,\cdot)}$ **then**
  $X \xleftarrow{\$} \{X : (T, X, Y) \in \mathcal{L}\}$
else if $X \in \mathrm{Dom}_{\mathcal{L}(T,\cdot)}$ **then**
  bad ← true
  $\boxed{X \xleftarrow{\$} \Sigma^n \setminus \mathrm{Dom}_{\mathcal{L}(T,\cdot)}}$
end if
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(T, X, Y)\}$
return X

Figure 18: Pseudocode for a ±TRP (boxed statements included) and for a ±TRF (boxed statements excluded). Caching logic is explicitly included in both cases. ±TRPs and ±TRFs are identical-until-bad, and we exploit this in Lemma 7 (Strong TPRP-TPRF switch) to show that they are birthday indistinguishable. $\mathcal{L}$ is initialised to the empty set. We define $\mathrm{Ran}_{\mathcal{L}(T,\cdot)} = \{Y : (T, X, Y) \in \mathcal{L}\}$ and $\mathrm{Dom}_{\mathcal{L}(T,\cdot)} = \{X : (T, X, Y) \in \mathcal{L}\}$.

Lemma 7 (Strong TPRP-TPRF switching lemma (adapted from [HR03]))**.** *Let π be a TRP with domain Σ^n, and let F be a TRF with the same domain. Let A be an adversary that may make pointless queries. Suppose A makes queries on at most R tweaks, and on each tweak makes at most q queries, and that A makes at most f queries overall. Then*

$$\overset{\Delta}{\underset{A,q}{}}\begin{pmatrix}\pm\pi\\ \pm F\end{pmatrix} \leq R \cdot \frac{q(q-1)}{2^{n+1}} \leq R \cdot \frac{q^2}{2^{n+1}}$$

and

$$\overset{\Delta}{\underset{A,f}{}}\begin{pmatrix}\pm\pi\\ \pm F\end{pmatrix} \leq \frac{f(f-1)}{2^{n+1}} \leq \frac{f^2}{2^{n+1}}.$$

Proof. As before, we appeal to Lemma 2 (Adversaries have no need to make pointless queries) and WLOG only consider adversaries that do not make pointless queries. Consider Figure 18. When boxed statements are not included, the pseudocode implements F and F^{-1} via lazy sampling. When boxed statements are included, the pseudocode implements π and π^{-1}. Therefore $\pm F$ and $\pm\pi$ are identical-until-bad. Examining the probability of setting bad on each query gives the result. □

Lemma 8 (Multi-user advantage to single-user advantage - adapted from [BCK96, Lemma 3.2])**.** *Let $\mathcal{O}, \mathcal{P}$ be oracles. Let A be an adversary making at most $R \times q$ queries to the multi-user oracles $R \times \mathcal{O}$ or $R \times \mathcal{P}$. Let s be the maximum computational cost of simulating either of $\mathcal{O}$ or $\mathcal{P}$. Then there is an explicit reduction $\mathcal{R}$ making at most q queries and with computational cost $t + (R-1)qs$ such that*

$$\mathop{\Delta}_{A,\substack{R\times q\\ t}}\left(\begin{matrix} R\times\mathcal{O} \\ R\times\mathcal{P}\end{matrix}\right) = R \cdot \mathop{\Delta}_{\mathcal{R}(A),q\atop t+(R-1)qs}\left(\begin{matrix}\mathcal{O}\\ \mathcal{P}\end{matrix}\right).$$

Proof. $\mathcal{R}(A)$ does the following. $\mathcal{R}(A)$ is given access to a single oracle $\mathcal{X}$, which is either an instance of $\mathcal{O}$ or $\mathcal{P}$. First, $\mathcal{R}(A)$ chooses an index $1 \le I \le R$ uniformly at random. $\mathcal{R}(A)$ runs A on oracles $(\mathcal{X}_1, \ldots \mathcal{X}_r)$ that $\mathcal{R}(A)$ simulates as

$$\mathcal{X}_j = \begin{cases} \mathcal{P}_j & j < I \\ \mathcal{X} & j = I \\ \mathcal{O}_j & j > I \end{cases}$$

where each $\mathcal{O}_j$ and $\mathcal{P}_j$ are simulated independent copies of the oracles $\mathcal{O}$ and $\mathcal{P}$. When A terminates by outputting a bit, $\mathcal{R}(A)$ forwards this bit as its own output. $\mathcal{R}(A)$ runs in at most $t + (R-1)qs$ units of work. This is because $\mathcal{R}(A)$ simulates the q queries A makes to each of the oracles in $\mathcal{X}_j$, each taking s units of work whenever $j \neq I$. Also, $\mathcal{R}(A)$ makes at most p queries to $\mathcal{X}$. By the law of total probability, the advantage of $\mathcal{R}(A)$ is bounded by the mean advantage of A against several different constructions:

$$\begin{aligned}
\mathop{\Delta}_{\mathcal{R}(A),q\atop t+(R-1)qs}\left(\begin{matrix}\mathcal{O}\\ \mathcal{P}\end{matrix}\right) &= \left|\mathbb{P}\left[\mathcal{R}(A)^{\mathcal{O}} \Rightarrow 1\right] - \mathbb{P}\left[\mathcal{R}(A)^{\mathcal{P}} \Rightarrow 1\right]\right| \\
&= \left|\sum_i \mathbb{P}\left[\mathcal{R}(A)^{\mathcal{O}} \Rightarrow 1 \;\middle|\; I = i\right]\mathbb{P}\left[I = i\right] \;-\; \mathbb{P}\left[\mathcal{R}(A)^{\mathcal{P}} \Rightarrow 1 \;\middle|\; I = i\right]\mathbb{P}\left[I = i\right]\right| \\
&= \frac{1}{R}\left|\sum_i \mathbb{P}\left[\mathcal{R}(A)^{\mathcal{O}} \Rightarrow 1 \;\middle|\; I = i\right] \;-\; \mathbb{P}\left[\mathcal{R}(A)^{\mathcal{P}} \Rightarrow 1 \;\middle|\; I = i\right]\right| \\
&= \frac{1}{R}\Big| \quad \mathbb{P}[A^{\mathcal{O}_1,\mathcal{O}_2,\ldots,\mathcal{O}_R} \Rightarrow 1] - \mathbb{P}[A^{\mathcal{P}_1,\mathcal{O}_2,\ldots,\mathcal{O}_R} \Rightarrow 1] \\
&\qquad + \quad \mathbb{P}[A^{\mathcal{P}_1,\mathcal{O}_2,\ldots,\mathcal{O}_R} \Rightarrow 1] - \mathbb{P}[A^{\mathcal{P}_1,\mathcal{P}_2,\ldots,\mathcal{O}_R} \Rightarrow 1] \\
&\qquad + \quad \ldots \\
&\qquad + \quad \mathbb{P}[A^{\mathcal{P}_1,\mathcal{P}_2,\ldots,\mathcal{O}_R} \Rightarrow 1] - \mathbb{P}[A^{\mathcal{P}_1,\mathcal{P}_2,\ldots,\mathcal{P}_R} \Rightarrow 1] \quad \Big| \\
&= \frac{1}{R}\Big| \quad \mathbb{P}[A^{\mathcal{O}_1,\mathcal{O}_2,\ldots,\mathcal{O}_R} \Rightarrow 1] - \mathbb{P}[A^{\mathcal{P}_1,\mathcal{P}_2,\ldots,\mathcal{P}_R} \Rightarrow 1] \quad \Big| \\
&= \frac{1}{R}\mathop{\Delta}_{A,\substack{R\times q\\ t}}\left(\begin{matrix} R\times\mathcal{O} \\ R\times\mathcal{P}\end{matrix}\right)
\end{aligned}$$

Multiplying both sides by R gives the result. □

A.2 Security of CTR mode

This appendix gives supporting proofs for the results in Section 4.2.1.

Lemma 9 (±TPRF security of CTR)**.** *Let π_{ctr} be an RP, and let F be a length-preserving TRF. Let ℓ be the maximum allowable message length in blocks. Let A be an adversary that is extremely nonce-respecting, in the sense that A never queries any pair of (nonce, message) inputs (I, X) and (I', X') to either oracle where the sequences $I, I+1, \ldots, I+(\ell-1)$ and $I', I'+1, \ldots, I'+(\ell-1)$ overlap. Suppose that A can make at most q queries. Then we have that*

$$\mathop{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{CTR}\left[\pi_{ctr}\right]\\ \pm F\end{matrix}\right) \le \frac{\ell^2 q^2}{2^{n+1}}.$$

Proof. First, we apply Corollary 5 (replacing suboracles) to swap π_{ctr} for an RF, G.

$$\mathbf{\Delta}_{A,q,\ell}\begin{pmatrix}\pm\,\mathrm{CTR}\,[+\pi_{ctr}]\\ \pm F\end{pmatrix} \le \mathbf{\Delta}_{\mathcal{R}(A),\ell q}\begin{pmatrix}+\pi_{ctr}\\ +G\end{pmatrix} + \mathbf{\Delta}_{A,q,\ell}\begin{pmatrix}\pm\,\mathrm{CTR}\,[+G]\\ \pm F\end{pmatrix}$$

The first term is bounded by Lemma 7 (TPRP-TPRF switching lemma):

$$\le \frac{\ell^2 q^2}{2^{n+1}} + \mathbf{\Delta}_{A,q,\ell}\begin{pmatrix}\pm\,\mathrm{CTR}\,[+G]\\ \pm F\end{pmatrix}$$

It remains to show that

$$\mathbf{\Delta}_{A,\ell}\begin{pmatrix}\pm\,\mathrm{CTR}\,[+G]\\ \pm F\end{pmatrix} = 0.$$

Since A is extremely nonce respecting, each input to $+G$ is unique. So the output of each query to $+G$ is a fresh uniform random value. So, each oracle output of $\pm\,\mathrm{CTR}$ is the (possibly truncated) concatenation of fresh uniform random values, and so must itself be fresh uniform random. Thus, when answering extremely nonce-respecting queries $\pm\,\mathrm{CTR}\,[+G]$ exactly simulates a random function, and so the advantage must be zero as required. □

A.3 GLEVIAN-HASH has the ε-AXU$_2$ property

This appendix gives supporting proofs for the results in Section 4.2.2. Throughout this section, let $K_{\mathcal{H}1}$ be a uniform random variable over $GF(2^n)$, let $\pi_{\mathcal{H}2}$ be an RP on Σ^n, let F be an RF on Σ^n and $\oplus$ denotes addition in $GF(2^n)$. Further, let $(U,V) \neq (U',V')$ be a pair of arbitrary unequal inputs. In all lemmas except[20] Lemma 12, let (U,V) have total integer-block length at most $\ell < 2^n - 1$ (that is, the sum of the integer-block lengths of U and V is at most ℓ, following the convention laid out in Section 4.1.1), so that $\mathsf{pad}(U,V)$ has length at most ℓ blocks, and likewise for (U',V'). Let $d \in \Sigma^n$ be an arbitrary output difference.

In this appendix, we show that GLEVIAN-HASH has the ε-AXU$_2$ property with $\varepsilon = 5\ell^2/2^{n+1}$. We begin with a lemma that will allow us to swap the RP π for an RF F without significantly affecting ε.

Lemma 35 (GLEVIAN-HASH behaves similarly when instantiated with similar subcomponents)**.** *Let Π, Ψ be arbitrary oracles. There exists an adversary A such that A makes at most ℓ queries and*

$$\begin{aligned}&|\mathbb{P}\left[\text{GVN-HASH}_{K_{\mathcal{H}1}}[\Pi](U,V) \;\oplus\; \text{GVN-HASH}_{K_{\mathcal{H}1}}[\Pi](U',V') = d\right]\\ &\quad - \quad \mathbb{P}\left[\text{GVN-HASH}_{K_{\mathcal{H}1}}[\Psi](U,V) \;\oplus\; \text{GVN-HASH}_{K_{\mathcal{H}1}}[\Psi](U',V') = d\right]| = \mathbf{\Delta}_{A,\ell}\begin{pmatrix}\Pi\\ \Psi\end{pmatrix}. \qquad (21)\end{aligned}$$

where the probability is taken over Π, Ψ and $K_{\mathcal{H}1}$.

Proof. Define A to be an adversary that acts on challenge oracle $\mathcal{O}$ as follows. A generates a random $K_{\mathcal{H}1}$, and then uses $\mathcal{O}$ and $K_{\mathcal{H}1}$ to calculate whether

$$\text{GVN-HASH}_{K_{\mathcal{H}1}}[\mathcal{O}](U,V) \;\oplus\; \text{GVN-HASH}_{K_{\mathcal{H}1}}[\mathcal{O}](U',V') = d$$

and outputs 1 if so, and 0 otherwise. Then the advantage that A achieves is exactly the LHS of equality (21), as required. □

Next, we show that the probability of the RP ever processing the same input block at different indices is negligible, owing to the masking by $\alpha^i K_{\mathcal{H}1}$.

Lemma 36 (Block cipher inputs collide with small probability)**.** *Let* coll *be the event that a pair of blocks with different indices result in a repeated input to GLEVIAN-HASH's subcomponent; that is, define the event* coll *as follows:*

$$\mathsf{coll} = \bigcup_{\substack{B,B' \in \mathsf{pad}(U,V)\cup\mathsf{pad}(U',V')\\ \mathsf{index}(B)\neq\mathsf{index}(B')}} \left[\alpha^{\mathsf{index}(B)} K_{\mathcal{H}1} \oplus B = \alpha^{\mathsf{index}(B')} K_{\mathcal{H}1} \oplus B'\right]$$

Then

$$\mathbb{P}[\mathsf{coll}] \le \frac{2\ell(2\ell-1)}{2^{n+1}}.$$

where the probability is taken over $K_{\mathcal{H}1}$.

[20] Lemma 12 is vacuously true if $\ell \geq 2^n - 1$, and so we may drop the requirement there.

Proof. By a union bound

$$\mathbb{P}\left[\mathsf{coll}\right] \leq \sum_{\substack{B,B' \in \mathsf{pad}(U,V) \cup \mathsf{pad}(U',V') \\ \mathsf{index}(B) \neq \mathsf{index}(B')}} \mathbb{P}\left[\alpha^{\mathsf{index}(B)} K_{\mathcal{H}1} \oplus B = \alpha^{\mathsf{index}(B')} K_{\mathcal{H}1} \oplus B'\right].$$

For each B, B', we have that $\mathsf{index}(B) \neq \mathsf{index}(B')$. Since α is a primitive element, we must have that $\alpha^{\mathsf{index}(B)} \neq \alpha^{\mathsf{index}(B')}$. So we can re-arrange:

$$\begin{aligned}
&= \sum_{\substack{B,B' \in \mathsf{pad}(U,V) \cup \mathsf{pad}(U',V') \\ \mathsf{index}(B) \neq \mathsf{index}(B')}} \mathbb{P}\left[K_{\mathcal{H}1} = \frac{B \oplus B'}{\alpha^{\mathsf{index}(B)} \oplus \alpha^{\mathsf{index}(B')}}\right] \\
&= \sum_{\substack{B,B' \in \mathsf{pad}(U,V) \cup \mathsf{pad}(U',V') \\ \mathsf{index}(B) \neq \mathsf{index}(B')}} 2^{-n} \\
&\leq \frac{(2\ell)(2\ell-1)}{2} \cdot \frac{1}{2^n}
\end{aligned}$$

noting that $\mathsf{pad}(U,V) \cup \mathsf{pad}(U',V')$ consists of at most 2ℓ blocks.

□

Lemma 37 (Conditioned on $\overline{\mathsf{coll}}$, GLEVIAN-HASH with the RF F has the ε-AXU$_2$ property)**.** *Let* coll *be defined as in Lemma 36. Then*

$$\mathbb{P}\left[\text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U,V) \oplus \text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U',V') = d \mid \overline{\mathsf{coll}}\right] = \frac{1}{2^n}$$

where the probability is taken over F and $K_{\mathcal{H}1}$.

Proof. Recall that

$$\text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U,V) \oplus \text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U',V') = \sum_{B \in \mathsf{pad}(U,V) \cup \mathsf{pad}(U',V')} F\left(\alpha^{\mathsf{index}(B)} K_{\mathcal{H}1} \oplus B\right).$$

The output of this summation will be uniformly distributed unless we can partition the sum into pairs where in each pair the inputs to F are equal, in which case the sum will be zero. Since $(U,V) \neq (U',V')$, any such pairing cannot happen solely between blocks of the same index. Unless coll occurs, this pairing up cannot happen between blocks of different indices. □

Lemma 12 (GLEVIAN-HASH has the ε-AXU$_2$ property)**.** *Let $\pi_{\mathcal{H}2}$ be an RP, and let $K_{\mathcal{H}1}$ be a uniform random variable over $GF(2^n)$. Let $(U,V) \neq (U',V')$ be a pair of inputs, each of total block length at most $\ell \geq 1$ blocks (where lengths are as defined in Section 4.1.1). Let d be an arbitrary output difference. Then*

$$\mathbb{P}[\text{GVN-HASH}_{K_{\mathcal{H}1}}[\pi_{\mathcal{H}2}](U,V) \oplus \text{GVN-HASH}_{K_{\mathcal{H}1}}[\pi_{\mathcal{H}2}](U',V') = d] \leq \frac{5\ell^2}{2^{n+1}}.$$

where the probability is taken over the choice of $\pi_{\mathcal{H}2}$ and $K_{\mathcal{H}1}$.

Proof. By Lemma 35 (GLEVIAN-HASH behaves similarly when instantiated with similar subcomponents), we have that (introducing the adversary A defined by that lemma):

$$\begin{aligned}
&\mathbb{P}[\text{GVN-HASH}_{K_{\mathcal{H}1}}[\pi_{\mathcal{H}2}](U,V) \oplus \text{GVN-HASH}_{K_{\mathcal{H}1}}[\pi_{\mathcal{H}2}](U',V') = d] \leq \\
&\qquad \mathbb{P}[\text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U,V) \oplus \text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U',V') = d] + \mathbf{Adv}_{A,\ell}\begin{pmatrix}\pi_{\mathcal{H}2} \\ F\end{pmatrix}
\end{aligned}$$

We then apply Lemma 7 (Strong TPRP-TPRF switch) to the second term, and the law of total probability to the first term, to obtain:

$$\leq \mathbb{P}\left[\text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U,V) \oplus \text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U',V') = d \mid \mathsf{coll}\right] \mathbb{P}\left[\mathsf{coll}\right]$$

$$+ \mathbb{P}\left[\text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U,V) \oplus \text{GVN-HASH}_{K_{\mathcal{H}1}}[F](U',V') = d \mid \overline{\mathsf{coll}}\right] \mathbb{P}\left[\overline{\mathsf{coll}}\right]$$
$$+ \frac{\ell(\ell-1)}{2^{n+1}}$$

The first term is at most $\mathbb{P}[\mathsf{coll}]$ which is bounded by Lemma 36. The second is bounded by Lemma 37, so we obtain:

$$\leq \frac{2\ell(2\ell-1)}{2^{n+1}} + \frac{1}{2^n} + \frac{\ell(\ell-1)}{2^{n+1}} = \frac{5\ell^2 - 3\ell + 2}{2^{n+1}} \leq \frac{5\ell^2}{2^{n+1}} \qquad \text{since } \ell \geq 1.$$

□

A.4 VIGORNIAN-HASH has the ε-AXU$_2$ property

This appendix gives supporting proofs for the results in Section 4.2.3.

Lemma 38 ((U,V) is uniquely encoded by polynomial $\mathsf{poly}_{U,V}(x)$)**.** *Let U,V,U',V' be bitstrings of length less than 2^n, and suppose that $\mathsf{poly}_{U,V}(x) = \mathsf{poly}_{U',V'}(x)$. Then $(U,V) = (U',V')$.*

Proof. Let U,V,U',V' be bitstrings, and suppose that $\mathsf{poly}_{U,V}(x) = \mathsf{poly}_{U',V'}(x)$. Then $\mathsf{poly}_{U,V}(x)$ and $\mathsf{poly}_{U',V'}(x)$ are equal in their x^1 coefficient, which is defined to be $len(V)$ and $len(V')$ respectively, and so we may conclude that $len(V) = len(V')$. Call this value v. Further, $\mathsf{poly}_{U,V}(x)$ and $\mathsf{poly}_{U',V'}(x)$ are equal in their $x^{2+\lceil v/n \rceil}$ coefficient, which is defined to be $len(U)$ and $len(U')$ respectively, and so we may conclude that $len(U) = len(U')$. Call this value u. Now we have that V is encoded in the coefficients of $x^2, \ldots, x^{1+\lceil v/n \rceil}$ of $\mathsf{poly}_{U,V}(x)$, and similarly V' is encoded in the same coefficients of $\mathsf{poly}_{U',V'}(x)$. Since this encoding is injective for bitstrings V and V' of the same length, we may conclude that $V = V'$. Similarly, we may conclude that $U = U'$. □

Lemma 13 (VIGORNIAN-HASH has the ε-AXU$_2$ property)**.** *Let $K_{\mathcal{H}}$ be a uniform random variable over $GF(2^n)$. Let $(U,V) \neq (U',V')$ be a pair of inputs, each of total length at most $\ell \geq 1$ blocks (where lengths are as defined in Section 4.1). Let d be an arbitrary output difference. Then*

$$\mathbb{P}[\text{VGN-HASH}_{K_{\mathcal{H}}}(U,V) \oplus \text{VGN-HASH}_{K_{\mathcal{H}}}(U',V') = d] \leq \frac{(\ell+2)}{2^n},$$

where the probability is taken over the choice of $K_{\mathcal{H}}$.

Proof. By Lemma 38, we may conclude $\mathsf{poly}_{U,V}(x) \neq \mathsf{poly}_{U',V'}(x)$.

VGN-HASH$_{K_{\mathcal{H}}}(U,V) \oplus$ VGN-HASH$_{K_{\mathcal{H}}}(U',V') = d$ if and only if $K_{\mathcal{H}}$ is a root of the polynomial $g(x) = \mathsf{poly}_{U,V}(x) + \mathsf{poly}_{U',V'}(x) + d$.

Note that $len(\mathsf{pad}(U,V)) = len(U) + len(V) + 2$. Consequently the degree of $g(x)$ is at most $\ell + 2$.

Since neither $\mathsf{poly}_{U,V}(x)$ nor $\mathsf{poly}_{U',V'}(x)$ has a constant term, we may conclude the degree of $g(x)$ is greater than zero, and so it has at most $\ell + 2$ roots. Therefore

$$\mathbb{P}[\text{VGN-HASH}_{K_{\mathcal{H}}}(U,V) \oplus \text{VGN-HASH}_{K_{\mathcal{H}}}(U',V') = d] \leq \frac{\ell+2}{2^n}$$

as required. □

A.5 Security of the LRW2 construction

This appendix gives supporting proofs for the results in Section 4.2.4. Appealing to Lemma 2 (Adversaries have no need to make pointless queries), in this section WLOG we only consider adversaries that do not make pointless queries. Further, we count queries starting from number 0.

The LRW2 construction was originally proved secure in [LRW02, Theorem 2]. Our proof incorporates ideas from the original as well as competing work from Minematsu [Min07]. Our proof achieves the stronger of the two bounds from the competing works. Further, we aim to make our proof as simple as possible. We work from the pseudocode in Figure 7, which demonstrate identical-until-bad implementations of LRW2 and a TRP.

Algorithm 12 $\mathcal{O}_1(T, M)$ $\boxed{\mathcal{O}_2(T, M)}$

1: $C \xleftarrow{\$} \Sigma^n \setminus \mathrm{Im}\,(\Pi(T, \cdot))$
2: $N \leftarrow M \oplus \mathcal{H}(T)$
3: $B \leftarrow C \oplus \mathcal{H}(T)$
4: **if** $N \in \mathrm{Dom}(\pi_{lrw2})$ **then**
5: bad $\leftarrow$ true
6: $\boxed{B \leftarrow \pi_{lrw2}(N)}$
7: $\boxed{C \leftarrow \mathcal{H}(T) \oplus B}$
8: **else if** $B \in \mathrm{Im}(\pi_{lrw2})$ **then**
9: bad $\leftarrow$ true
10: $\boxed{B \xleftarrow{\$} \Sigma^n \setminus \mathrm{Im}(\pi_{lrw2})}$
11: $\boxed{C \leftarrow \mathcal{H}(T) \oplus B}$
12: **end if**
13: $\pi_{lrw2}(N) \leftarrow B$
14: $\Pi(T, M) \leftarrow C$
15: **return** C

Algorithm 13 $\mathcal{O}_1^{-1}(T, M)$ $\boxed{\mathcal{O}_2^{-1}(T, M)}$

1: $M \xleftarrow{\$} \Sigma^n \setminus \mathrm{Dom}\,(\Pi(T, \cdot))$
2: $N \leftarrow M \oplus \mathcal{H}(T)$
3: $B \leftarrow C \oplus \mathcal{H}(T)$
4: **if** $B \in \mathrm{Im}(\pi_{lrw2})$ **then**
5: bad $\leftarrow$ true
6: $\boxed{N \leftarrow \pi_{lrw2}^{-1}(B)}$
7: $\boxed{M \leftarrow \mathcal{H}(T) \oplus N}$
8: **else if** $N \in \mathrm{Dom}(\pi_{lrw2})$ **then**
9: bad $\leftarrow$ true
10: $\boxed{N \xleftarrow{\$} \Sigma^n \setminus \mathrm{Dom}(\pi_{lrw2})}$
11: $\boxed{M \leftarrow \mathcal{H}(T) \oplus N}$
12: **end if**
13: $\pi_{lrw2}(N) \leftarrow B$
14: $\Pi(T, M) \leftarrow C$
15: **return** M

Figure 7: Pseudocode jointly defining the $\mathrm{LRW2}\,[\mathcal{H}, \pi_{lrw2}]$ construction and a TRP Π using lazy sampling. $\mathcal{O}_1$, which does not include the boxed statements, implements a TRP. $\mathcal{O}_2$, which includes the boxed statements, implements LRW2 (see Lemma 39). Given a query of the form (T, M), LRW2 encrypts M to C using tweak T. The shared state of $\mathcal{O}_x$ and $\mathcal{O}_x^{-1}$ is initialised so that bad is set to false and π_{lrw2} and Π are the empty map. For brevity, we do not include caching logic to deal with pointless queries.

Lemma 39 (LRW2 pseudocode is correct)**.** *Including boxed statements, and assuming pointless queries are not made, the algorithms in Figure 7 correctly simulate the* $\pm\,\mathrm{LRW2}\,[\pi_{lrw2}, \mathcal{H}]$ *construction.*

Proof. We argue that the following invariants are preserved on every query:

1. The cache π_{lrw2} exactly implements an RP via lazy sampling
2. For all T, $\mathrm{Im}(\Pi(T, \cdot)) \subseteq \mathrm{Im}(\pi_{lrw2}) + \mathcal{H}(T)$
3. For all T, $\mathrm{Dom}(\Pi(T, \cdot)) \subseteq \mathrm{Dom}(\pi_{lrw2}) + \mathcal{H}(T)$

The algorithms in Figure 7 answer queries in such a way that $C = \mathcal{H}(T) \oplus \pi_{lrw2}\,(\mathcal{H}(T) \oplus M)$ as specified by the $\mathrm{LRW2}[\pi_{lrw2}, \mathcal{H}]$ construction, and so invariant (1) is sufficient to prove the lemma. For encryption queries, we use invariant (2) to support our argument that invariant (1) is preserved. For decryption queries, we use invariant (3) to support our argument that invariant (1) is preserved.

WLOG, we show that making a query to the encrypt interface preserves invariants (1), (2), and (3). The decrypt argument is similar. Suppose the encrypt interface has been queried with (T, M), and invariants (1), (2), and (3) hold. Suppose further that $M \notin \mathrm{Dom}(\Pi(T, \cdot))$, i.e. (T, M) is not a pointless query. Let N, B, and C be as calculated in the encryption algorithm of Figure 7.

Firstly, we show that invariant (2) is preserved. On each encryption query, the output C will be the only addition to the set $\mathrm{Im}(\Pi(T, \cdot))$. Therefore we must show that $C \in \mathrm{Im}(\pi_{lrw2}) + \mathcal{H}(T)$. But this is the case, because $C = B + \mathcal{H}(T)$, and note that $B \in \mathrm{Im}(\pi_{lrw2})$ because $B = \pi_{lrw2}(N)$. Invariant (3) is preserved by a similar argument.

Finally, we show that invariant (1) is preserved. If $N \in \mathrm{Dom}(\pi_{lrw2})$, then invariant (1) is preserved since π_{lrw2} is not updated. By contrast, if $N \notin \mathrm{Dom}(\pi_{lrw2})$, then the algorithm (once simplified) performs the following steps to calculate B:

(a) Samples $B \leftarrow^{\$} \Sigma^n \setminus \mathrm{Im}(\Pi(T, \cdot)) + \mathcal{H}(T)$,

(b) If $B \in \mathrm{Im}(\pi_{lrw2})$, resamples $B \leftarrow^{\$} \Sigma^n \setminus \mathrm{Im}(\pi_{lrw2})$

To show invariant (1), we must show that this is consistent with lazy-sampling π_{lrw2} as an RP; i.e. we must show that it is equivalent to directly sampling $B \xleftarrow{\$} \Sigma^n \setminus \mathrm{Im}(\pi_{lrw2})$. Since step (b) correctly resamples any values that were sampled from $\mathrm{Im}(\pi_{lrw2})$ in step (a), it remains only to show that all values of $\Sigma^n \setminus \mathrm{Im}(\pi_{lrw2})$ are possible outcomes of step (a), i.e. to show that

$$\Sigma^n \setminus \mathrm{Im}(\pi_{lrw2}) \subseteq (\Sigma^n \setminus \mathrm{Im}(\Pi(T,\cdot))) + \mathcal{H}(T).$$

But this exactly follows from invariant (2); we have that

$$\mathrm{Im}(\Pi(T,\cdot)) \subseteq \mathrm{Im}(\pi_{lrw2}) + \mathcal{H}(T)$$

and so

$$\mathrm{Im}(\Pi(T,\cdot) + \mathcal{H}(T)) \subseteq \mathrm{Im}(\pi_{lrw2})$$

and so

$$\Sigma^n \setminus \mathrm{Im}(\pi_{lrw2}) \subseteq \Sigma^n \setminus \mathrm{Im}(\Pi(T,\cdot) + \mathcal{H}(T))$$

and so

$$\Sigma^n \setminus \mathrm{Im}(\pi_{lrw2}) \subseteq (\Sigma^n \setminus \mathrm{Im}(\Pi(T,\cdot))) + \mathcal{H}(T).$$

as required.

□

Lemma 14 (Bound on probability that adversary sets bad on query i to LRW2)**.** *Let A be an adversary asking queries with tweaks of length at most ℓ blocks, and let bad_i be the event that the code of Figure 7 reaches line 5 or 9 (i.e. the lines that set bad to true) on their i^{th} query (beginning the count from 0). Then*

$$\mathbb{P}[\mathsf{bad}_i] \leq 2i\varepsilon_\ell$$

Proof. In $\mathcal{O}_1$, since π_{lrw2} and $\mathcal{H}$ are independent, oracle outputs do not depend on $\mathcal{H}$, and so the calculation of bad_i and all interaction with $\mathcal{H}$ can be deferred until the end. Once all queries have been made, the bad_i is triggered exactly when there is a repeated N or B. That is, for bad_i to be set we need $\mathcal{H}(T_i) \oplus \mathcal{H}(T_j) \in \{M_i + M_j, C_i + C_j\}$ for some $j < i$. But since $\mathcal{H}$ is ε-AXU$_2$, this happens with probability (over $\mathcal{H}$) at most $2\varepsilon_\ell$ for each j, and thus (by a union bound) with overall probability at most $2i\varepsilon_\ell$ overall. □

Lemma 15 (Bound on probability that adversary has set bad after q queries to LRW2)**.** *Let A be an adversary asking queries of with tweaks of length at most ℓ blocks. The probability that* bad *is set after q queries to $\pm\mathcal{O}_1$ as defined in Figure 7 is bounded by*

$$\mathbb{P}[\mathsf{bad}] \leq q(q-1)\varepsilon_\ell$$

More generally, in the event that the adversary has access to R copies of $\pm\mathcal{O}_1$ with q queries to each, the probability that bad *is set is bounded by*

$$\mathbb{P}[\mathsf{bad}] \leq Rq(q-1)\varepsilon_\ell$$

Proof. For the single oracle case, we apply a union bound, and then Lemma 14 (Bound on probability that adversary sets bad on query i to LRW2).

$$\begin{aligned}
&\mathbb{P}[\mathsf{bad}] \\
&\leq \sum_i \mathbb{P}[\mathsf{bad}_i] \\
&\leq \sum_i 2i\varepsilon_\ell \\
&= q(q-1)\varepsilon_\ell
\end{aligned}$$

For the multiple oracle case, we apply a union bound in turn to this equation. □

Lemma 16 ($\pm$TPRP security of LRW2)**.** *Let π_{lrw2} be an RP, and let $\mathcal{H}$ be ε-AXU$_2$ for some $\varepsilon = \varepsilon_\ell$. Then provided π_{lrw2} and $\mathcal{H}$ are independent, the distinguishing advantage between* LRW2 *and a TRP Π with message space Σ^n and tweak space Σ^* against an adversary A making at most q queries where each tweak has length at most ℓ blocks is bounded by*

$$\mathop{\Delta}_{A,q}\left(\begin{matrix}\pm\,\mathrm{LRW2}\,[\pi_{lrw2},\mathcal{H}] \\ \pm\Pi\end{matrix}\right) \leq q(q-1)\varepsilon_\ell.$$

Proof. Consider identical-until-bad oracles $\pm\mathcal{O}_1$ and $\pm\mathcal{O}_2$ as defined in Figure 7. Since $\mathcal{O}_1$ does not include the boxed statements, lines 2-12 have no effect on the output values of $\pm\mathcal{O}_1$, or on the Π cache. Thus $\mathcal{O}_1$ precisely implements a $\pm TRP$ via lazy sampling. Conversely, Lemma 39 shows that $\pm\mathcal{O}_2$ exactly implements $\pm\,\mathrm{LRW2}\,[\pi_{lrw2}, \mathcal{H}]$.

Directly from the definition of $\mathcal{O}_x$, we have that $\mathcal{O}_1$ and $\mathcal{O}_2$ are identical-until-bad, since boxed lines only occur on lines directly following $\mathsf{bad} \leftarrow \mathsf{true}$.

Thus we can invoke Lemma 6 (Fundamental lemma of game-playing) and then Lemma 15 (Bound on probability that adversary has set bad after q queries to LRW2) to find that

$$\begin{aligned}
&\overset{\Delta}{}_{A,q}\left(\begin{matrix}\pm\,\mathrm{LRW2}\,[\pi_{lrw2}, \mathcal{H}] \\ \pm\Pi\end{matrix}\right) \\
&\leq \mathbb{P}\,[\mathsf{bad}] \\
&= q(q-1)\varepsilon_\ell
\end{aligned}$$

□

A.6 Security of PIV construction

This appendix gives supporting proofs for the results in Section 4.2.5. Appealing to Lemma 2 (Adversaries have no need to make pointless queries), throughout this section we assume WLOG that A does not make pointless queries. Note that we continue to assume A does not make pointless queries even as we progress the proof and modify the game that A is playing in ways that mean such queries no longer seem pointless. This does not affect the correctness of the proof.

Lemma 40. *Let A be an adversary with query constraint q, where the length after padding of X or Y on each query is at most ℓ blocks. Let Ψ be a TRP and F be a TRF, both wide-tweak wide-block; let Π be a TRP and G be a TRF, both wide-tweak narrow-block. Then there is an explicit reduction $\mathcal{R}$ such that $\mathcal{R}(A)$ is a (possibly nonce-disrespecting) adversary making at most $2q$ queries to $\mathcal{B}$, with the tweak of each query of length at most ℓ, and*

$$\overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm\mathcal{B}, \pm\mathcal{C}] \\ \pm\Psi\end{matrix}\right) \leq \overset{\Delta}{}_{\mathcal{R}(A),2q,\ell}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) \quad + \quad \overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm G, \pm\mathcal{C}] \\ \pm F\end{matrix}\right) \quad + \quad \frac{5q^2}{2^{n+1}}.$$

Proof.

$$\begin{aligned}
&\overbrace{\overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm\mathcal{B}, \pm\mathcal{C}] \\ \pm\Psi\end{matrix}\right)}^{\text{Apply Lemma 3, Triangle, swapping } \pm\Psi \text{ to } \pm F} \\
&\leq \overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm\mathcal{B}, \pm\mathcal{C}] \\ \pm F\end{matrix}\right) && + \overbrace{\overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm F \\ \pm\Psi\end{matrix}\right)}^{\text{Apply Lemma 7, TPRP-TPRF}} \\
&\leq \overbrace{\overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm\mathcal{B}, \pm\mathcal{C}] \\ \pm F\end{matrix}\right)}^{\text{Apply Corollary 5, Replacing suboracles}} && + \frac{q^2}{2^{n+1}} \\
&\leq \overset{\Delta}{}_{\mathcal{R}(A),2q,\ell}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) && + \overbrace{\overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm\Pi, \pm\mathcal{C}] \\ \pm F\end{matrix}\right)}^{\text{Apply Corollary 5, Replacing suboracles}} && + \frac{q^2}{2^{n+1}} \\
&\leq \overset{\Delta}{}_{\mathcal{R}(A),2q,\ell}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) && + \overbrace{\overset{\Delta}{}_{\mathcal{S}(A),2q,\ell}\left(\begin{matrix}\pm\Pi \\ \pm G\end{matrix}\right)}^{\text{Apply Lemma 7, TPRP-TPRF}} && + \overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm G, \pm\mathcal{C}] \\ \pm F\end{matrix}\right) && + \frac{q^2}{2^{n+1}} \\
&\leq \overset{\Delta}{}_{\mathcal{R}(A),2q,\ell}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) && + \frac{(2q)^2}{2^{n+1}} && + \overset{\Delta}{}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm G, \pm\mathcal{C}] \\ \pm F\end{matrix}\right) && + \frac{q^2}{2^{n+1}}
\end{aligned}$$

Collecting terms completes the proof. □

To conclude the security of the PIV construction, it remains to find a bound on

$$\mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm G,\pm\mathcal{C}]\\ \pm F\end{matrix}\right).$$

Note that it is not possible to immediately substitute $\mathcal{C}$ for an ideal primitive in the same way as we did for $\mathcal{B}$, because our PIV security lemma will only require $\mathcal{C}$ to be secure against extremely nonce respecting adversaries - that is, adversaries that never query with a nonce within ℓ of a nonce previously sent to any oracle.

±TPRF security of PIV construction We use the techniques of lazy sampling and game-hopping to approach this task. We define a chain of oracles.

The first of these, $\mathcal{O}_1$, implements the $\pm\,\mathrm{PIV}\,[\pm G,\pm\mathcal{C}]$ exactly. In the definition of $\mathcal{O}_1$, each query results in two calls to $\pm G$. Lazy sampling code (including caching code) for $\pm G$ has been inlined, using bad events to ensure correctness. Note that our code may send pointless queries to $\mathcal{C}$ (which we assume will have its own caching mechanism). Caching code for the overall construction is not included; it is not necessary because A does not make pointless queries. Our code is given in Algorithms 14 and 15.

$\mathcal{O}_2$, defined in the same Algorithms, differs from the real PIV in $\mathcal{O}_1$ in that it (a) doesn't cache queries to G; and (b) enforces that $\mathcal{C}$ is never queried with neighbouring nonces. Note that $\mathcal{O}_2$ respects $\mathcal{C}$'s nonce requirement, since line 15 ensures that all nonces passed to $\mathcal{C}$ are not within ℓ of any other nonce. We define $\mathcal{O}_3$ to be equal to $\mathcal{O}_2$ except with $\mathcal{C}$ replaced by a TRF H.

Lemma 41. *Let A be an adversary with query constraint q. Let F be a wide-tweak wide-block TRF, and let H be a narrow-tweak wide-block TRF. Then there exists an efficient and extremely nonce-respecting reduction $\mathcal{S}$ such that*

$$\mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm G,\pm\mathcal{C}]\\ \pm F\end{matrix}\right) \le \mathbb{P}[A^{\pm\mathcal{O}_2}\ \textit{sets}\ \mathsf{bad}] + \mathbf{\Delta}_{\mathcal{S}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C}\\ \pm H\end{matrix}\right).$$

Proof.

$$\overbrace{\mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm G,\pm\mathcal{C}]\\ \pm F\end{matrix}\right)}^{\text{Apply Lemma 3, Triangle}}$$

$$\le \overbrace{\mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\,\mathrm{PIV}\,[\pm G,\pm\mathcal{C}]\\ \pm\mathcal{O}_1\end{matrix}\right)}^{\text{zero by direct simulation}} + \mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_1\\ \pm\mathcal{O}_2\end{matrix}\right) + \mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_2\\ \pm\mathcal{O}_3\end{matrix}\right) + \mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_3\\ \pm F\end{matrix}\right)$$

$$\le 0 + \overbrace{\mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_1\\ \pm\mathcal{O}_2\end{matrix}\right)}^{\text{Apply identical until } \mathsf{bad} = \mathsf{bad}_1 \cup \mathsf{bad}_2 \cup \mathsf{bad}_3} + \mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_2\\ \pm\mathcal{O}_3\end{matrix}\right) + \mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_3\\ \pm F\end{matrix}\right)$$

$$\le 0 + \mathbb{P}[A^{\pm\mathcal{O}_2}\ \text{sets}\ \mathsf{bad}] + \overbrace{\mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_2\\ \pm\mathcal{O}_3\end{matrix}\right)}^{\text{Apply Lemma 4, Replacing suboracles}} + \mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_3\\ \pm F\end{matrix}\right)$$

Note that our invocation of Lemma 4 in this step introduces a reduction $\mathcal{S}(A)$ which is extremely nonce-respecting.

$$\le 0 + \mathbb{P}[A^{\pm\mathcal{O}_2}\ \text{sets}\ \mathsf{bad}] + \mathbf{\Delta}_{\mathcal{S}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C}\\ H\end{matrix}\right) + \overbrace{\mathbf{\Delta}_{A,q,\ell}\left(\begin{matrix}\pm\mathcal{O}_3\\ \pm F\end{matrix}\right)}^{\text{Zero by equivalence of oracles}}$$

Algorithm 14 $\mathcal{O}(U, X)$:

1: $j \leftarrow j + 1$
2: $(X_L, X_R) \leftarrow X$
3:
4: $T \leftarrow (U, X_L)$
5: $IV \xleftarrow{\$} \Sigma^n$
6: **if** $(T, X_R) \in \operatorname{dom} G$ **then**
7: $\quad \mathsf{bad}_1 \leftarrow \mathsf{true}$
8: $\quad \boxed{IV \xleftarrow{\$} \{IV : (T, X_R, IV) \in G\}}$
9: **end if**
10: $G \leftarrow G \cup \{(T, X_R, IV)\}$
11:
12: $N_j \leftarrow IV$
13: **if** $N_j \in \cup_{i<j} \{N_i + k : -\ell < k < \ell\}$ **then**
14: $\quad \mathsf{bad}_2 \leftarrow \mathsf{true}$
15: $\quad \boxed{N_j \xleftarrow{\$} \Sigma^n \setminus \bigcup_{i<j} \{N_i + k : -\ell < k < \ell\}}$
16: **end if**
17: $Y_L \leftarrow \mathcal{C}(N_j, X_L)$
18:
19: $T' \leftarrow (U, Y_L)$
20: $Y_R \xleftarrow{\$} \Sigma^n$
21: **if** $(T', IV) \in \operatorname{dom} G$ **then**
22: $\quad \mathsf{bad}_3 \leftarrow \mathsf{true}$
23: $\quad \boxed{Y_R \xleftarrow{\$} \{Y_R : (T', IV, Y_R) \in G\}}$
24: **end if**
25: $G \leftarrow G \cup \{(T', IV, Y_R)\}$
26: **return** $Y_L \parallel Y_R$

Algorithm 15 $\mathcal{O}^{-1}(U, Y)$:

1: $j \leftarrow j + 1$
2: $(Y_L, Y_R) \leftarrow Y$
3:
4: $T \leftarrow (U, Y_L)$
5: $IV \xleftarrow{\$} \Sigma^n$
6: **if** $(T, Y_R) \in \operatorname{range} G$ **then**
7: $\quad \mathsf{bad}_1 \leftarrow \mathsf{true}$
8: $\quad \boxed{IV \xleftarrow{\$} \{IV : (T, IV, Y_R) \in G\}}$
9: **end if**
10: $G \leftarrow G \cup \{(T, IV, Y_R)\}$
11:
12: $N_j \leftarrow IV$
13: **if** $N_j \in \cup_{i<j} \{N_i + k : -\ell < k < \ell\}$ **then**
14: $\quad \mathsf{bad}_2 \leftarrow \mathsf{true}$
15: $\quad \boxed{N_j \xleftarrow{\$} \Sigma^n \setminus \bigcup_{i<j} \{N_i + k : -\ell < k < \ell\}}$
16: **end if**
17: $X_L \leftarrow \mathcal{C}^{-1}(N_j, Y_L)$
18:
19: $T' \leftarrow (U, X_L)$
20: $X_R \xleftarrow{\$} \Sigma^n$
21: **if** $(T', IV) \in \operatorname{range} G$ **then**
22: $\quad \mathsf{bad}_3 \leftarrow \mathsf{true}$
23: $\quad \boxed{X_R \xleftarrow{\$} \{X_R : (T', X_R, IV)\}}$
24: **end if**
25: $G \leftarrow G \cup \{(T', X_R, IV)\}$
26: **return** $X_L \parallel X_R$

Figure 19: Pseudocode definitions of $\mathcal{O}_1$ (implements $\pm\mathrm{PIV}[\pm G, \pm\mathcal{C}]$) and $\mathcal{O}_2$ (implements an intermediate oracle) and $\mathcal{O}_3$. $\mathcal{O}_1$ includes the boxed statements but not dashed boxed statements, and vice versa for $\mathcal{O}_2$. We define $\mathcal{O}_3$ (implements $\pm F$) to be equal to $\mathcal{O}_2$ except with $\mathcal{C}$ replaced with a TRF H. j is initialised to 0, and G is initialised to empty. Note that when N_j is sampled on line 15, this set can be empty, but only when $q(2\ell + 1) > 2^n$. At that point our advantage bound is ≥ 1 anyway, so this does not affect the result.

Since A does not make pointless queries, queries to F in the fourth term always result in responses that are freshly sampled uniform random values; meanwhile $\pm\mathcal{O}_3$ also always outputs freshly sampled uniform random values, and so it is a perfect implementation of F.

$$\leq \quad 0 \quad + \quad \mathbb{P}[A^{\pm\mathcal{O}_2} \text{ sets } \mathsf{bad}] \quad + \quad \mathcal{S}^{\Delta}_{(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C}\\ \pm H\end{matrix}\right) + \quad 0$$

□

It remains to find a bound on $\mathbb{P}[A^{\pm\mathcal{O}_2} \text{ sets } \mathsf{bad}]$. On the face of it, bounding $\mathbb{P}[A^{\pm\mathcal{O}_2} \text{ sets } \mathsf{bad}]$ seems difficult because we have no guarantee that $\mathcal{C}$ does not perform pathological behaviour that enables A to learn how to set bad. However, $\mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad}]$ is much easier to bound, as $\mathcal{C}$ has been replaced by a TRF. Further, accounting for the amount this probability may differ from the one needed is also possible, as we can reduce the difference to the security of $\mathcal{C}$. Made concrete,

Lemma 42. *Let A be an adversary. Then there is some reduction $\mathcal{T}$ such that $\mathcal{T}(A)$ is extremely nonce-respecting and*

$$\mathbb{P}[A^{\pm\mathcal{O}_2} \textit{ sets } \mathsf{bad}] \leq \mathcal{T}^{\Delta}_{(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C}\\ \pm H\end{matrix}\right) + \mathbb{P}[A^{\pm\mathcal{O}_3} \textit{ sets } \mathsf{bad}].$$

Proof. First, observe that

$$\mathbb{P}[A^{\pm\mathcal{O}_2} \text{ sets } \mathsf{bad}] \leq \left|\mathbb{P}[A^{\pm\mathcal{O}_2} \text{ sets } \mathsf{bad}] - \mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad}]\right| + \mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad}].$$

We define $\mathcal{T}(A)$ as follows: $\mathcal{T}(A)$ runs A and simulates A's $\pm\mathcal{O}_{2 \text{ or } 3}$ oracles, forwarding any queries to $\mathcal{C}$ or H to its challenge oracles, and emits 1 iff bad is set. $\mathcal{T}(A)$'s advantage in distinguishing $\pm\mathcal{C}$ and $\pm H$ is exactly the difference in the probabilities that A sets bad in oracles 2 and 3, and so

$$\left|\mathbb{P}[A^{\pm\mathcal{O}_2} \text{ sets } \mathsf{bad}] - \mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad}]\right| = \mathcal{T}(A),q,\ell \begin{pmatrix} \pm\mathcal{C} \\ \pm H \end{pmatrix}$$

Moreover, we note that by definition of $\mathcal{O}_2$, $\mathcal{T}(A)$ is extremely nonce respecting as required. □

Note that since $\mathcal{T}(A)$ is an extremely nonce respecting adversary for $\mathcal{C}$, when $\mathcal{C} = \text{CTR}$ we will have that Lemma 9 (CTR security) applies. Next, we need the following result.

Lemma 43. *Let A be an adversary that makes at most q queries of length at most ℓ. Then*

$$\mathbb{P}[A^{\pm\mathcal{O}_3} \textit{ sets } \mathsf{bad}] \leq \frac{(\ell+4)q^2}{2^{n+1}}.$$

Proof. The bound is vacuously true unless $1 \leq q \leq 2^{n-1}$, which we assume WLOG. By a union bound,

$$\mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad}] \leq \sum_j \mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad} \text{ on query } j] \leq \sum_j \sum_i \mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad}_i \text{ on query } j].$$

We will deal with each summand separately. For simplicity, we consider an adversary that is making an encryption query; the case where A makes a decryption query is symmetric.

$\mathbb{P}[A^{\pm\mathcal{O}_3}$ **sets** bad_1 **on query** $j]$**.** First, we consider the probability that A sets bad_1 on query j. In order to trigger bad_1, A must make a query that specifies an X and U value such that $T = (U, X_L)$ and $(T, X_R) \in \operatorname{dom} G$.

Entries to $\operatorname{dom}(G(T,\cdot))$ must have originated from one of four possible lines of the pseudocode for $\mathcal{O}_3$. Firstly, on lines 10 of the encrypt algorithm and 25 of the decrypt algorithm, A can choose values which are added to $\operatorname{dom}(G(T,\cdot))$. However, A cannot trigger bad_1 by querying one of these values, because doing so would involve making a pointless query. Secondly, on lines 25 of the encrypt algorithm and 10 of the decrypt algorithm, an IV is added to $\operatorname{dom}(G(T,\cdot))$. When interacting with $\mathcal{O}_3$, A has no information about the IV values because all outputs are uniform random and independent of the IVs (because $\mathcal{C}$ has been replaced with a TRF), and because the IVs are chosen uniformly at random. Therefore A's queries can be considered independent of the IVs. On query j, once T is fixed, from the adversary's perspective there are at least $2^n - j$ possibilities for X_R that might trigger bad_1, of which at most j actually will (depending on what the IV was), and so A's probability of triggering bad_1 is at most

$$\frac{j}{2^n - j}.$$

$\mathbb{P}[A^{\pm\mathcal{O}_3}$ **sets** bad_2 **on query** $j]$. On line 12 we have that $N_j \leftarrow IV$, which was uniformly chosen from 2^n values. Since query i has length at most ℓ blocks, there are at most $2\ell - 1$ possible choices for N_j that would cause it to overlap with the choice for N_i. Thus the if statement is entered with probability at most

$$\frac{1}{2^n} \sum_{i<j} 2\ell - 1 = \frac{j(2\ell-1)}{2^{n+1}}.$$

$\mathbb{P}[A^{\pm\mathcal{O}_3}$ **sets** bad_3 **on query** $j]$. bad_3 is set only if IV, which is chosen uniformly from 2^n values, is one of at most $2j$ values. So we have the probability that bad_3 is set on $2j$ values is at most

$$\frac{2j}{2^n}.$$

Pulling these subresults together,

$$\begin{aligned}
\mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad}] &\leq \sum_{j=0}^{q-1} \left(\frac{j}{2^n - j} + \frac{j(2\ell-1)}{2^{n+1}} + \frac{2j}{2^n} \right) \qquad \text{(note that } j < q \leq 2^{n-1}\text{, so } \tfrac{j}{2^n - j} \leq \tfrac{j}{2^{n-1}}\text{)} \\
&\leq \sum_{j=0}^{q-1} \left(\frac{j}{2^{n-1}} + \frac{j(2\ell-1)}{2^{n+1}} + \frac{j}{2^{n-1}} \right) \\
&= \sum_{j=0}^{q-1} \frac{j}{2^{n+1}} (4 + 2\ell - 1 + 4) \\
&= \frac{(2\ell+7)}{2^{n+1}} \sum_{j=0}^{q-1} j \leq \frac{(2\ell+8)}{2^{n+1}} \frac{q(q-1)}{2} \\
&\leq \frac{(\ell+4)q^2}{2^{n+1}}.
\end{aligned}$$

□

Finally, we may now bring together our bound on the security of PIV mode, concluding Appendix A.6 and proving Lemma 17:

Lemma 17 (±TPRP security of PIV)**.** *Let A be an adversary that makes at most q queries, where the length after padding of X, Y and U on each query is at most ℓ blocks. Let Ψ be a wide-tweak wide-block TRP, Π be a wide-tweak, narrow-block TRP, and H be a narrow-nonce wide-block TRF. Then we can exhibit a reduction $\mathcal{R}(A)$ and extremely nonce-respecting reductions $\mathcal{S}(A)$ and $\mathcal{T}(A)$, such that $\mathcal{R}(A)$ makes at most $2q$ queries to $\mathcal{B}$ with tweaks of length at most ℓ blocks, $\mathcal{S}(A)$ and $\mathcal{T}(A)$ each make at most q queries of length $\ell - 1$ blocks to $\mathcal{C}$, and*

$$\mathbf{Adv}^{\pm\text{prp}}_{A,q,\ell}\left(\begin{matrix}\pm\,\text{PIV}\,[\pm\mathcal{B}, \pm\mathcal{C}] \\ \pm\Psi\end{matrix}\right) \leq \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{R}(A),2q}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) + \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{S}(A),q,\ell-1}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{T}(A),q,\ell-1}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \frac{(\ell+9)q^2}{2^{n+1}}.$$

where $\mathcal{R}(A)$ asks queries with tweaks of length at most ℓ blocks.

Proof.

$$\begin{aligned}
&\overbrace{\mathbf{Adv}^{\pm\text{prp}}_{A,q,\ell}\left(\begin{matrix}\pm\,\text{PIV}\,[\pm\mathcal{B}, \pm\mathcal{C}] \\ \pm\Psi\end{matrix}\right)}^{\text{Apply Lemma 40}} \\
&\quad\leq \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{R}(A),2q}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) + \overbrace{\mathbf{Adv}^{\pm\text{prp}}_{A,q,\ell}\left(\begin{matrix}\pm\,\text{PIV}\,[\pm G, \pm\mathcal{C}] \\ \pm F\end{matrix}\right)}^{\text{Apply Lemma 41}} + \frac{5q^2}{2^{n+1}} \\
&\quad\leq \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{R}(A),2q}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) + \overbrace{\mathbb{P}[A^{\pm\mathcal{O}_2} \text{ sets } \mathsf{bad}]}^{\text{Apply Lemma 42}} + \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{S}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \frac{5q^2}{2^{n+1}} \\
&\quad\leq \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{R}(A),2q}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) + \left(\mathbf{Adv}^{\pm\text{prp}}_{\mathcal{T}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \overbrace{\mathbb{P}[A^{\pm\mathcal{O}_3} \text{ sets } \mathsf{bad}]}^{\text{Apply Lemma 43}} \right) + \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{S}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \frac{5q^2}{2^{n+1}} \\
&\quad\leq \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{R}(A),2q}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) + \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{S}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{T}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \frac{(\ell+4)q^2}{2^{n+1}} + \frac{5q^2}{2^{n+1}} \\
&\quad\leq \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{R}(A),2q}\left(\begin{matrix}\pm\mathcal{B} \\ \pm\Pi\end{matrix}\right) + \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{S}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \mathbf{Adv}^{\pm\text{prp}}_{\mathcal{T}(A),q,\ell}\left(\begin{matrix}\pm\mathcal{C} \\ \pm H\end{matrix}\right) + \frac{(\ell+9)q^2}{2^{n+1}}.
\end{aligned}$$

□

Corollary 18 (±TPRP security of WTBC)**.** *Let A be an adversary that makes at most q queries, where the length after padding of encryption and decryption queries, and of each tweak, is at most ℓ*

blocks. Let Ψ be a wide-tweak wide-block TRP, let LRW2 *be instantiated with an RP π_{lrw2}, let $\mathcal{H}$ be ε-AXU_2 for some $\varepsilon = \varepsilon_\ell$ and let* CTR *be instantiated with RP π_{ctr}. Then*

$$\mathbf{Adv}^{\pm\mathrm{WTBC}}_{A,q,\ell}(\pm\Psi) \le 4q^2\varepsilon_\ell + \frac{(\ell+2)^2q^2}{2^n}.$$

Proof. We apply Lemma 17 (±TPRP security of PIV), and instantiate the various components. Let A be an adversary. Let Π be a wide-tweak, narrow-block TRP, and H be a narrow-nonce wide-block TRF. Then there is a reduction $\mathcal{R}$ and extremely nonce-respecting reductions $\mathcal{S}$ and $\mathcal{T}$, such that

$$\overbrace{\mathbf{Adv}^{\pm\mathrm{WTBC}}_{A,q,\ell}(\pm\Psi)}^{\text{Expand definition}}$$

$$= \overbrace{\mathbf{Adv}^{\pm\mathrm{PIV}[\pm\mathrm{LRW2}[\pi_{lrw2},\mathcal{H}],\mathrm{CTR}[\pi_{ctr}]]}_{A,q,\ell}(\pm\Psi)}^{\text{Apply Lemma 17, ±TPRP security of PIV}}$$

$$\le \overbrace{\mathbf{Adv}^{\pm\mathrm{LRW2}[\pi_{lrw2},\mathcal{H}]}_{\mathcal{R}(A),2q,\ell}(\pm\Pi)}^{\text{Apply Lemma 16, ±TPRP security of LRW2}} + \overbrace{\mathbf{Adv}^{\mathrm{CTR}[\pi_{ctr}]}_{\mathcal{S}(A),q,\ell-1}(\pm H) + \mathbf{Adv}^{\mathrm{CTR}[\pi_{ctr}]}_{\mathcal{T}(A),q,\ell-1}(\pm H)}^{\text{Apply Lemma 9, ±TPRF security of CTR}} + \frac{(\ell+9)q^2}{2^{n+1}}$$

$$\le 2q(2q-1)\varepsilon_\ell + \frac{2(\ell-1)^2q^2}{2^{n+1}} + \frac{(\ell+9)q^2}{2^{n+1}} = 2q(2q-1)\varepsilon_\ell + \overbrace{\frac{(2\ell^2-3\ell+11)q^2}{2^{n+1}}}^{\text{Apply that } \ell \ge 1}$$

$$\le 4q^2\varepsilon_\ell + \frac{2(\ell+2)^2q^2}{2^{n+1}}$$

which suffices. □

A.7 Security of PTE1 construction

This appendix gives supporting proofs for the results in Section 4.2.6.

Lemma 19 (DAE security of PTE1 $[\Psi]$ in the RUP model follows from ±TPRP security of Ψ)**.** *Let A be an adversary making at most q queries each consisting of at most ℓ blocks. Let Ψ be a (not necessarily ideal) wide block cipher, and let Π be an ideal TRP. Let* PTE1 *provide an encrypt, decrypt and leakage interface as defined in Figure 11. Let F be a wide-tweak wide-block stretch-n TRF and let G be a wide-tweak wide-block shrink-n TRF. Then there is a reduction $\mathcal{R}$ such that*

$$\mathbf{Adv}^{\pm\mathrm{PTE1}[\Psi]}_{A,q,\ell}(F,\perp,G) \le \mathbf{Adv}^{\pm\Psi}_{\mathcal{R}(A),q,\ell+1}(\pm\Pi) + \frac{q^2}{2^n}.$$

Proof. In the following, H is a wide-tweak wide-block TRF (with no stretch).

$$\overbrace{\mathbf{Adv}^{\pm\mathrm{PTE1}[\Psi]}_{A,q,\ell}(F,\perp,G)}^{\text{Triangle inequality}} \le \overbrace{\mathbf{Adv}^{\pm\mathrm{PTE1}[\Psi]}_{A,q,\ell}(\pm\mathrm{PTE1}[H])}^{\text{Lemma 4 (Replacing suboracles)}} + \mathbf{Adv}^{\pm\mathrm{PTE1}[H]}_{A,q,\ell}(F,\perp,G)$$

$$\le \overbrace{\mathbf{Adv}^{\pm\Psi}_{\mathcal{R}(A),q,\ell+1}(\pm H)}^{\text{Triangle inequality}} + \mathbf{Adv}^{\pm\mathrm{PTE1}[H]}_{A,q,\ell}(F,\perp,G)$$

$$\le \mathbf{Adv}^{\pm\Psi}_{\mathcal{R}(A),q,\ell+1}(\pm\Pi) + \overbrace{\mathbf{Adv}^{\pm\Pi}_{\mathcal{R}(A),q,\ell+1}(\pm H)}^{\text{Lemma 7 (TPRP-TPRF switch)}} + \overbrace{\mathbf{Adv}^{\pm\mathrm{PTE1}[H]}_{A,q,\ell}(F,\perp,G)}^{\text{Direct reasoning}}$$

$$\le \mathbf{Adv}^{\pm\Psi}_{\mathcal{R}(A),q,\ell+1}(\pm\Pi) + \frac{q(q-1)}{2^{n+1}} + \frac{q}{2^n}$$

To arrive at the final line, we note that as $\mathcal{R}(A)$ simulates PTE1 for A it always asks queries of at least n bits in length, and we observe that the only way to distinguish $\underset{\sim}{\pm}\,\mathrm{PTE1}[H]$ from $(F, \perp, G)$ is to make a successful decryption query, which occurs independently with probability 2^{-n} on each query.

The last two terms are bounded by $\frac{q^2}{2^n}$, so the advantage is at most

$$\mathbf{Adv}^{\pm\Psi}_{\mathcal{R}(A),q,\ell+1}(\pm\Pi) + \frac{q^2}{2^n}$$

as required. □

Lemma 20 (DAE security of LD-DAE in the RUP model)**.** *Let π_{lrw2}, π_{ctr} be ideal block ciphers and let $\mathcal{H}$ be an ε-AXU_2 for some $\varepsilon = \varepsilon_\ell$. Write* LD-DAE *for* LD-DAE$[\mathcal{H}, \pi_{lrw2}, \pi_{ctr}]$ *and let its leakage interface be as defined in Algorithm 6 (that is, when* $\sim$LD-DAE *is queried with (A, C, T) it returns all but the last n bits of* $\mathrm{WTBC}^{-1}(A, C\|T)$ *whenever* $\text{LD-DAE}^{-1}(A, C, T) = \perp$*).*

Let F be a wide-tweak wide-block stretch-n TRF and let G be a wide-tweak wide-block shrink-n TRF. Let A be an adversary making at most q queries each consisting of at most $\ell \geq 1$ blocks.

Then

$$\mathbf{Adv}^{\underset{\sim}{\pm}\,\text{LD-DAE}}_{A,q,\ell}(F, \perp, G) \leq q^2\left(4\varepsilon_\ell + \frac{(\ell+4)^2}{2^n}\right)$$

Proof. Let Π be an ideal wide-tweak, wide-block cipher. Recalling that LD-DAE $=$ PTE1 [WTBC],

$$\begin{aligned}
\overbrace{\mathbf{Adv}^{\underset{\sim}{\pm}\,\text{LD-DAE}}_{A,q,\ell}(F, \perp, G)}^{\text{Apply definition of LD-DAE}} &= \overbrace{\mathbf{Adv}^{\underset{\sim}{\pm}\,\mathrm{PTE1}[\mathrm{WTBC}]}_{A,q,\ell}(F, \perp, G)}^{\text{Apply Lemma 19 (PTE1 security)}} \leq \overbrace{\mathbf{Adv}^{\pm\,\mathrm{WTBC}}_{\mathcal{R}(A),q,\ell+1}(\pm\Pi)}^{\substack{\text{Apply Corollary 18,}\\ \pm\text{TPRP security of WTBC}}} + \frac{q^2}{2^n} \\
&\leq \left(4q^2\varepsilon_\ell + \frac{(\ell+2)^2q^2}{2^n}\right) + \frac{q^2}{2^n} \\
&\leq 4q^2\varepsilon_\ell + \frac{(\ell+4)^2q^2}{2^n}
\end{aligned}$$

□

A.8 Security of NKD construction

This appendix gives supporting proofs for the results in Section 4.2.8. Throughout this appendix, recall from Section 4.1.2 that an adversary making (R, q, f) or $(R \times q, f)$ queries against a possibly multi-user oracle that provides an encrypt, decrypt and leakage interface can make $Rq + 2f$ queries in total, consisting of q encryption queries to each of either R nonces or users; f decryption queries to any combination of nonces/users; and f leakage queries to any combination of nonces/users.

Lemma 44 (NKD with an ideal KDF is identical to copies of the underlying DAE scheme)**.** *Let Π be a (not necessarily ideal) stretch-n DAE scheme with key space Σ^k for some k. Let F, F' be wide-tweak wide-block stretch-n TRFs, G, G' be wide-tweak wide-block shrink-n TRFs, and H be an RF with codomain Σ^k. Let the leakage interface of Π be arbitrary and let the leakage interface of* NKD $[\Pi, H]$ *queried with (N, A, C, T) return $\sim\Pi_{G(N)}(A, C, T)$ as defined in Algorithm 9.*

Let A be an adversary with computational resource bounded by t and with query constraints (R, q, f). Then there is a reduction $\mathcal{R}$ with computational resource bounded by $(Rq + 2f)s_T$ where s_T is the cost of a lookup into a table of size at most $R + 2f$, such that

$$\mathbf{Adv}^{\underset{\sim}{\pm}\,\mathrm{NKD}[\Pi,H]}_{A,R,q,f,\ell}(F, \perp, G) = \mathbf{Adv}^{\underset{\sim}{\pm}\Pi}_{\mathcal{R}(A),(R\times q),f,\ell}\begin{pmatrix}(R+2f)\times\\ (R+2f)\times(F', \perp, G')\end{pmatrix}$$

Proof. $\mathcal{R}(A)$ runs A and for each query A makes, $\mathcal{R}(A)$ looks up whether it is under a fresh nonce (at a cost of s_T). If it is not then $\mathcal{R}(A)$ forwards the query to the same oracle as it used the last time it saw the nonce. Otherwise, $\mathcal{R}(A)$ forwards the query to a fresh oracle and stores the association between this nonce and the new oracle. In either case, $\mathcal{R}(A)$ simulates the oracles on the LHS perfectly for A as each novel nonce on the LHS generates a uniform random key in Σ^k, which is the same as using a fresh oracle in the multi-user oracle on the RHS. A uses at most $R + 2f$ distinct nonces in its queries and so $\mathcal{R}(A)$ requires access to at most $R + 2f$ oracles within the multi-user oracle on the RHS, and $\mathcal{R}(A)$ must perform exactly one lookup for each query that A makes. □

Lemma 22 (NKD security reduces to security of the KDF and the underlying DAE scheme)**.** *Let* Π *be a stretch-n DAE scheme with key space* Σ^k *for some* k*. Let* $\mathcal{K}$ *be a (not necessarily ideal) KDF scheme with codomain* Σ^k*. Let the leakage interface of* Π *be arbitrary and for any KDF* $\mathcal{K}'$ *let the leakage interface of* $\mathrm{NKD}[\Pi, \mathcal{K}']$ *queried with* (N, A, C, T) *return* $\sim\Pi_{\mathcal{K}'(N)}(A, C, T)$ *as in Algorithm 9.*

Let F, F' *be wide-tweak wide-block stretch-n TRFs and let* G, G' *be wide-tweak wide-block shrink-n TRFs. Let* H *be an RF with codomain* Σ^k*. Let* A *be an adversary with computational cost bounded by* t *and with query constraints* (R, q, f)*. Let* $s_{\Pi,\ell}$ *be the maximum computational cost of simulating a call to any of the interfaces of* Π *with inputs of length at most* ℓ *blocks and let* s_T *be the cost of a lookup into a table of size at most* $R + 2f$*.*

Then there are efficiently constructible reductions $\mathcal{R}, \mathcal{S}$ *such that*

$$\mathbf{Adv}^{\mathrm{mrae}}_{A,R,q,f,\ell}\left(\underset{\sim}{\pm}\,\mathrm{NKD}[\Pi,\mathcal{K}],\ F,\perp,G\right) \le \mathbf{Adv}^{\mathrm{prf}}_{\mathcal{R}(A),R+2f,t+(Rq+2f)s_{\Pi,\ell}}\left(\mathcal{K},H\right) + \mathbf{Adv}^{\mathrm{prf}}_{\mathcal{S}(A),(R\times q),f,\ell,t+(Rq+2f)s_T}\left(\begin{matrix}(R+2f)\times(\underset{\sim}{\pm}\Pi)\\(R+2f)\times(F',\perp,G')\end{matrix}\right)$$

Proof. Beginning with the security of NKD, apply the triangle inequality to isolate the KDF; then apply Lemma 4 (Replacing suboracles) with reduction $\mathcal{R}$. The cost per query of this reduction is no more than $s_{\Pi,\ell}$.

$$\begin{aligned}\mathbf{Adv}^{\mathrm{mrae}}_{A,R,q,f,\ell}\left(\underset{\sim}{\pm}\,\mathrm{NKD}[\Pi,\mathcal{K}],\ F,\perp,G\right) \le\ & \overbrace{\mathbf{Adv}^{\mathrm{mrae}}_{A,R,q,f,\ell}\left(\begin{matrix}\underset{\sim}{\pm}\,\mathrm{NKD}[\Pi,\mathcal{K}]\\ \underset{\sim}{\pm}\,\mathrm{NKD}[\Pi,H]\end{matrix}\right)}^{\text{Apply Lemma 4 (Replacing suboracles)}} + \mathbf{Adv}^{\mathrm{mrae}}_{A,R,q,f,\ell}\left(\underset{\sim}{\pm}\,\mathrm{NKD}[\Pi,H],\ F,\perp,G\right)\\ \le\ & \mathbf{Adv}^{\mathrm{prf}}_{\mathcal{R}(A),R+2f,t+(Rq+2f)s_{\Pi,\ell}}\left(\mathcal{K},H\right) + \mathbf{Adv}^{\mathrm{mrae}}_{A,R,q,f,\ell}\left(\underset{\sim}{\pm}\,\mathrm{NKD}[\Pi,H],\ F,\perp,G\right)\end{aligned}$$

Now apply Lemma 44 (NKD with an RF is identical to copies of Π) to the second term, where $\mathcal{S}$ is the reduction in that lemma, to obtain the result.

□

Lemma 23 (MRAE security of HD-AEAD from DAE security of LD-DAE)**.** *Let* $\mathcal{E}$ *be an efficient block cipher and let* $\pi, \pi_{\mathcal{H}2}, \pi_{lrw2}, \pi_{ctr}$ *be an RP. Let the leakage interface of* LD-DAE *be arbitrary and let* $\sim\mathrm{HD\text{-}AEAD}[\mathcal{H}, \mathcal{E}]$ *when queried with* (N, A, C, T) *be as in Algorithm 9, that is, it returns* $\sim\mathrm{LD\text{-}DAE}[\mathcal{H}[\mathcal{E}]_{(K_{\mathcal{H}})}, \mathcal{E}_{(K_{lrw2})}, \mathcal{E}_{(K_{ctr})}](A, C, T)$ *where* $K_{\mathcal{H}}, K_{lrw2}, K_{ctr}$ *are the output of* $\mathrm{XORP}_w[\mathcal{E}](N)$*.*

Let F, F' *be wide-tweak wide-block stretch-n TRFs and let* G, G' *be wide-tweak wide-block shrink-n TRFs.*

Let A *be an adversary with computational cost bounded by* t *and with query constraints* (R, q, f)*. Let* s_ℓ *be the computational cost of simulating a call to* $\underset{\sim}{\pm}\,\mathrm{HD\text{-}AEAD}[\mathcal{H}[\mathcal{E}], \mathcal{E}]$ *with inputs of length at most* ℓ *blocks, or the cost of a lookup into a table of size at most* $R + 2f$*, whichever is greater.*

Then there are efficiently constructible reductions $\mathcal{R}, \mathcal{S}_1, \ldots, \mathcal{S}_4$ *such that*

$$\mathbf{Adv}^{\mathrm{mrae}}_{A,R,q,f,\ell}\left(\underset{\sim}{\pm}\,\mathrm{HD\text{-}AEAD}[\mathcal{H}[\mathcal{E}],\mathcal{E}],\ F,\perp,G\right) \le (R+2f)\sum_{i=1}^{4}\mathbf{Adv}^{\mathrm{prp}}_{\mathcal{S}_i(A),t+3(Rq+2f)s_\ell}\left(\pm\mathcal{E},\pm\pi\right) \tag{4}$$

$$+\ \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}} \tag{5}$$

$$+\ \mathbf{Adv}^{\mathrm{prf}}_{\mathcal{R}(A),(R\times q),f,\ell}\left(\begin{matrix}(R+2f)\times(\underset{\sim}{\pm}\,\mathrm{LD\text{-}DAE}[\mathcal{H}[\pi_{\mathcal{H}2}],\pi_{lrw2},\pi_{ctr}])\\(R+2f)\times(F',\perp,G')\end{matrix}\right)$$

Note that in the computational advantage term here we do not give query constraints and implicitly allow the adversaries 2^n *queries; meanwhile, in the information theoretic term we do not give computational constraints and implicitly allow the adversary unbounded computation.*

Proof. Beginning with the MRAE security of HD-AEAD we unpack the definition of HD-AEAD and apply Lemma 22 (NKD security). H, $\mathcal{R}_1, \mathcal{R}_2$ are the RF and reductions in that lemma and by supposition $(Rq + 2f)s_\ell$ upper bounds the computational cost of those reductions.

$$\overbrace{\mathbf{Adv}^{\mathrm{mrae}}_{A,R,q,f,\ell}\left(\underset{\sim}{\pm}\,\mathrm{HD\text{-}AEAD}[\mathcal{H},\mathcal{E}],\ F,\perp,G\right)}^{\text{Apply Definition}} \tag{22}$$

$$
\begin{aligned}
&= \overbrace{\overset{\Delta}{}_{A,\overset{R,q,f,\ell}{t}}\left(\begin{matrix}\pm\!\!\!\sim \mathrm{NKD}\left[\text{LD-DAE}\left[\mathcal{H},\mathcal{E}\right],\mathrm{XORP}_w[\mathcal{E}_{(K)}]\right]\\ F,\perp,G\end{matrix}\right)}^{\text{Apply Lemma 22 (NKD security)}}\\
&\leq \overset{\Delta}{}_{\substack{\mathcal{R}_1(A),R+2f\\ t+(Rq+2f)s_\ell}}\left(\begin{matrix}\mathrm{XORP}_w[\mathcal{E}_{(K)}]\\ H\end{matrix}\right)\\
&\quad + \overbrace{\overset{\Delta}{}_{\substack{\mathcal{R}_2(A),(R\times q),f,\ell\\ t+(Rq+2f)s_\ell}}\left(\begin{matrix}(R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\mathcal{E}]_{(K_H)},\mathcal{E}_{(K_{lrw2})},\mathcal{E}_{(K_{ctr})}\right])\\ (R+2f)\times(F',\perp,G')\end{matrix}\right)}^{\text{Apply Lemma 3 (Triangle inequality)}}\\
&\leq \overset{\Delta}{}_{\substack{\mathcal{R}_1(A),R+2f\\ t+(Rq+2f)s_\ell}}\left(\begin{matrix}\mathrm{XORP}_w[\mathcal{E}_{(K)}]\\ H\end{matrix}\right) && (23)\\
&\quad + \overset{\Delta}{}_{\substack{\mathcal{R}_2(A),(R\times q),f,\ell\\ t+(Rq+2f)s_\ell}}\left(\begin{matrix}(R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\mathcal{E}]_{(K_H)},\mathcal{E}_{(K_{lrw2})},\mathcal{E}_{(K_{ctr})}\right])\\ (R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\pi],\pi_{lrw2},\pi_{ctr}\right])\end{matrix}\right) && (24)\\
&\quad + \overset{\Delta}{}_{\substack{\mathcal{R}_2(A),(R\times q),f,\ell}}\left(\begin{matrix}(R+2f)\times\left(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\pi],\pi_{lrw2},\pi_{ctr}\right]\right)\\ (R+2f)\times(F',\perp,G')\end{matrix}\right). && (25)
\end{aligned}
$$

We decompose the first two terms into expressions in terms of block cipher security (the last term, being information theoretic, admits no such decomposition). Recall from Section 4.1.5 that when describing the ±PRP advantage against a single-user block cipher oracle we sometimes omit the adversary's query limit, in which case we implicitly allow the adversary access to all 2^n input-output pairs (a permissive change).

For term (23), we use Corollary 5 (Replacing suboracles) and Lemma 21 (XORP security) where π_{xorp} is an RP. We can then efficiently construct an adversary that we denote $\mathcal{S}_{xorp}$ such that

$$
\overset{\Delta}{}_{\substack{\mathcal{R}_1(A),R+2f\\ t+(Rq+2f)s_\ell}}\left(\begin{matrix}\mathrm{XORP}_w[\mathcal{E}_{(K)}]\\ H\end{matrix}\right) \leq \overset{\Delta}{}_{\substack{\mathcal{S}_{xorp}(A)\\ t+(R(q+1)+4f)s_\ell}}\left(\begin{matrix}\mathcal{E}_{(K)}\\ \pi_{xorp}\end{matrix}\right) + \frac{(w+1)^4(R+2f)^3}{2^{2n+1}} + \frac{w(w+1)(R+2f)}{2^{n+1}}. \tag{26}
$$

Now decompose term (24) by Lemma 3 (Triangle inequality) and apply Lemma 4 (Replacing suboracles) to each of $\pi_{\mathcal{H}2}$, π_{lrw2} and π_{ctr} followed by Lemma 8 (Multi-user to single-user). We abuse notation slightly and write $\mathcal{E}_{(K_\mathcal{H})}$ for the block cipher component of $\mathcal{H}[\mathcal{E}]$ when $\mathcal{H}$ is keyed by $K_\mathcal{H}$; this is an abuse as we also use $K_\mathcal{H}$ to mean all the key material required for $\mathcal{H}$, which might be strictly more key material than required by $\mathcal{E}$. For concreteness, we denote the replacing-suboracles reductions (into which we subsume $\mathcal{R}_2$) by $\mathcal{S}_H, \mathcal{S}_{lrw2}, \mathcal{S}_{ctr}$, in (24). We likewise denote the multi-user to single-user reductions $\mathcal{T}_H, \mathcal{T}_{lrw2}, \mathcal{T}_{ctr}$ in (24) (where we subsume the corresponding replacing suboracles reduction). When computing computational bounds for the multi-user to single-user reductions below, note that we can simply count the queries $\mathcal{R}_2(A)$ makes to each LD-DAE oracle; if $\mathcal{R}_2(A)$ makes $Rq+2f$ LD-DAE queries (recall that an adversary can make f queries to each of the decrypt and leakage interfaces) then the computational cost of satisfying the block cipher calls must be no more than $(Rq+2f)s_\ell$, as s_ℓ incorporates the cost of the underlying block ciphers.

For (24) we then have:

$$
\begin{aligned}
(24) \quad &\leq \overbrace{\overset{\Delta}{}_{\substack{\mathcal{R}_2(A),(R\times q),f,\ell\\ t+(Rq+2f)s_\ell}}\left(\begin{matrix}(R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\mathcal{E}]_{(K_H)},\mathcal{E}_{(K_{lrw2})},\mathcal{E}_{(K_{ctr})}\right])\\ (R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\pi],\mathcal{E}_{(K_{lrw2})},\mathcal{E}_{(K_{ctr})}\right])\end{matrix}\right)}^{\text{Apply Lemma 4 (Replacing Suboracles)}}\\
&\quad + \overbrace{\overset{\Delta}{}_{\substack{\mathcal{R}_2(A),(R\times q),f,\ell\\ t+(Rq+2f)s_\ell}}\left(\begin{matrix}(R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\pi],\mathcal{E}_{(K_{lrw2})},\mathcal{E}_{(K_{ctr})}\right])\\ (R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\pi],\pi_{lrw2},\mathcal{E}_{(K_{ctr})}\right])\end{matrix}\right)}^{\text{Apply Lemma 4 (Replacing Suboracles)}}\\
&\quad + \overbrace{\overset{\Delta}{}_{\substack{\mathcal{R}_2(A),(R\times q),f,\ell\\ t+(Rq+2f)s_\ell}}\left(\begin{matrix}(R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\pi],\pi_{lrw2},\mathcal{E}_{(K_{ctr})}\right])\\ (R+2f)\times(\pm\!\!\!\sim \text{LD-DAE}\left[\mathcal{H}[\pi],\pi_{lrw2},\pi_{ctr}\right])\end{matrix}\right)}^{\text{Apply Lemma 4 (Replacing Suboracles)}}\\
&\leq \overbrace{\overset{\Delta}{}_{\substack{\mathcal{S}_H(A)\\ t+2(Rq+2f)s_\ell}}\left(\begin{matrix}(R+2f)\times\pm\mathcal{E}_{(K_\mathcal{H})}\\ (R+2f)\times\pm\pi_{\mathcal{H}2}\end{matrix}\right)}^{\text{Apply Lemma 8 (Multi-user to single-user)}} + \overbrace{\overset{\Delta}{}_{\substack{\mathcal{S}_{lrw2}(A)\\ t+2(Rq+2f)s_\ell}}\left(\begin{matrix}(R+2f)\times\pm\mathcal{E}_{(K_{lrw2})}\\ (R+2f)\times\pm\pi_{lrw2}\end{matrix}\right)}^{\text{Apply Lemma 8 (Multi-user to single-user)}}
\end{aligned}
$$

$$
\begin{aligned}
&\overbrace{+ \underset{t+2(Rq+2f)s_\ell}{\overset{\Delta}{\mathcal{S}_{ctr}(A)}} \begin{pmatrix} (R+2f) \times \pm\mathcal{E}_{(K_{ctr})} \\ (R+2f) \times \pm\pi_{ctr} \end{pmatrix}}^{\text{Apply Lemma 8 (Multi-user to single-user)}} \\
\leq\quad & (R+2f) \cdot \underset{t+3(Rq+2f)s_\ell}{\overset{\Delta}{\mathcal{T}_H(A)}} \begin{pmatrix} \pm\mathcal{E}_{(K_{\mathcal{H}})} \\ \pm\pi_{\mathcal{H}2} \end{pmatrix} + (R+2f) \cdot \underset{t+3(Rq+2f)s_\ell}{\overset{\Delta}{\mathcal{T}_{lrw2}(A)}} \begin{pmatrix} \pm\mathcal{E}_{(K_{lrw2})} \\ \pm\pi_{ctr} \end{pmatrix} \qquad (27) \\
& + (R+2f) \cdot \underset{t+3(Rq+2f)s_\ell}{\overset{\Delta}{\mathcal{T}_{ctr}(A)}} \begin{pmatrix} \pm\mathcal{E}_{(K_{ctr})} \\ \pm\pi_{ctr} \end{pmatrix}. \qquad (28)
\end{aligned}
$$

Note that each of the ±PRP adversaries in the decompositions of the XORP terms, (24) are trying to distinguish an instance of $\mathcal{E}$ with an independent uniform key from an independent RP; they are all playing the same game. Also note that the maximum computational cost among these ±PRP adversaries is $t + 3(Rq + 2f)s_\ell$. Thus we have shown that

$$(22) \leq (26) + (27) + (28) + (25)$$

Writing $\mathcal{R} = \mathcal{R}_2$ and the four reductions to ±PRP adversaries as $\mathcal{S}_1, \ldots, \mathcal{S}_4$ completes the proof. □

A.9 Security of HD-AEAD construction

This appendix gives supporting proofs for the results in Section 4.3.2.

Lemma 29 (Decryption queries at the end)**.** *Let A be an adversary playing G_{dist} between two multi-user worlds of the form $(R + f) \times (\pm\mathcal{O})$ and $(R + f) \times (+\mathcal{P}, \perp)$. Suppose that $\pm\mathcal{O}$ can be eagerly sampled, i.e. the output of each query is independent of previous query inputs. Then there exists an adversary $\mathcal{R}(A)$ which has the same query constraints as A, which is about as efficient to run and has advantage at least as large as A, and which makes all its encryption queries before any decryption queries.*

Proof. For $b \in \{0, 1\}$, construct the adversary $\mathcal{R}_b(A)$ as follows: $\mathcal{R}_b(A)$ runs A, forwards the encryption queries to $+\mathcal{O}$, but returns $\perp$ for decryption queries. When A has finished running, $\mathcal{R}_b(A)$ sends all of its decryption queries to $-\mathcal{O}$. If $\mathcal{R}_b(A)$ successfully forges, it outputs b; otherwise, it outputs whatever A returns. Note that $\mathcal{R}_b(A)$ successfully forges if and only if A successfully forges, since $\pm\mathcal{O}$ is eagerly sampleable.

We set $\mathcal{R}(A)$ to one of $\mathcal{R}_0(A), \mathcal{R}_1(A)$, choosing the one with maximal advantage. Then $\mathcal{R}(A)$ will have an advantage at least as big as A's. □

Note that in the previous lemma, although each $\mathcal{R}_b(A)$ is efficiently constructible by a human, the choice of $\mathcal{R}(A)$ may not be.

Lemma 30 (G_{dist} as the probability of forgery)**.** *Let A be an adversary playing G_{dist} between two multi-user worlds of the form $(R + f) \times (\pm\mathcal{O})$ and $(R + f) \times (+\mathcal{O}, \perp)$. Suppose that A makes all its encryption queries before its decryption queries. Then*

$$\underset{A}{\Delta}\left((R+f) \times \begin{pmatrix} +\mathcal{O}, & -\mathcal{O} \end{pmatrix}\right) \leq \mathbb{P}[A^{(R+f)\times(\pm\mathcal{O})} \text{ forges}].$$

Proof. Since A makes all of its encryption queries before all of its decryption queries, the results of all of the encryption queries must be the same, regardless of the behaviour of $\pm\mathcal{O}$. Therefore the two oracles are identical-until-bad, where bad is the event that A forges. Therefore by Lemma 6 (Fundamental lemma of game-playing), A's advantage in the distinguishing game is at most the probability of bad. (Note that we do need the caveat that A makes all of its encryption queries before its decryption queries, in case calls to $-\mathcal{O}$ affect the state of $+\mathcal{O}$.) □