Secure Archival is Hard... *Really* Hard

Christopher Smith
Stony Brook University
Stony Brook, New York
csasmith@cs.stonybrook.edu

Maliha Tabassum
Stony Brook University
Stony Brook, New York
matabassum@cs.stonybrook.edu

Soumya Chowdary Daruru
Stony Brook University
Stony Brook, New York
sdaruru@cs.stonybrook.edu

Gaurav Kulhare
Stony Brook University
Stony Brook, New York
gkulhare@cs.stonybrook.edu

Arvin Wang
Stony Brook University
Stony Brook, New York
arvwang@cs.stonybrook.edu

Ethan L. Miller
Pure Storage/ University of California, Santa Cruz
Santa Cruz, California
elm@ucsc.edu

Erez Zadok
Stony Brook University
Stony Brook, New York
ezk@cs.stonybrook.edu

Abstract

Archival systems are often tasked with storing highly valuable data that may be targeted by malicious actors. When the lifetime of the secret data is on the order of decades to centuries, the threat of improved cryptanalysis casts doubt on the long-term security of cryptographic techniques, which rely on hardness assumptions that are hard to prove over archival time scales. This threat makes the design of secure archival systems exceptionally difficult. Some archival systems turn a blind eye to this issue, hoping that current cryptographic techniques will not be broken; others often use techniques—such as secret sharing—that are impractical at scale. This position paper sheds light on the core challenges behind building practically viable secure long-term archives; we identify promising research avenues towards this goal.

CCS Concepts: • **Information systems → Digital libraries and archives**; • **Computer systems organization → Secondary storage organization**; • **Security and privacy → Database and storage security**; **Information-theoretic techniques**.

Keywords: Archival storage, encryption, secret-sharing, Harvest Now Decrypt Later, information-theoretic security

ACM Reference Format:
Christopher Smith, Maliha Tabassum, Soumya Chowdary Daruru, Gaurav Kulhare, Arvin Wang, Ethan L. Miller, and Erez Zadok. 2024. Secure Archival is Hard... *Really* Hard. In *16th ACM Workshop on Hot Topics in Storage and File Systems (HOTSTORAGE '24), July 8–9, 2024, Santa Clara, CA, USA.* ACM, New York, NY, USA, 9 pages. *https://doi.org/10.1145/3655038.3666093*

HOTSTORAGE '24, July 8–9, 2024, Santa Clara, CA, USA

ACM ISBN 979-8-4007-0630-1/24/07
https://doi.org/10.1145/3655038.3666093

1 Introduction

Archives seek to preserve information for the long-term. In this work we assume "long-term" to be on the order of a human lifetime or more. At a minimum, archives should be highly reliable and storage-efficient. Reliability means that user data is never lost, corrupted, or unavailable. Storage efficiency is critical because the cost of storage is proportional to the time and amount of data stored, and since archives accumulate data that is rarely deleted, the size of an archive could be on the order of multiple exabytes or more [29].

Archives often store highly valuable and/or sensitive data, such as public historical records, private medical data, or classified government secrets; these may be targets for adversaries willing to spend vast resources to attack the archive over the course of many years. Thus, archives should protect data security (*i.e.*, confidentiality, integrity, availability) with strong guarantees.

At a minimum, secure storage systems typically use encryption for data confidentiality, message authentication codes or signatures for integrity, and replication or erasure coding for availability. On the archival time scales, however, it becomes much more likely that cryptographic techniques used today will be broken in the future by cryptanalytic advances, and this event is impossible to rule out unless we resolve long-standing open questions like $\mathbb{P}$ vs. $\mathbb{NP}$. Naively re-encrypting data to use newer ciphers is also an inadequate approach because re-encrypting a large archive is prohibitively expensive (see Section 3). Furthermore, it fails to address the threat of adversaries who steal *encrypted* data now with the hopes of extracting useful information years down the line; this is called a "Harvest Now, Decrypt Later" attack—a threat being taken seriously by industry and government alike with the prospects of cryptographically viable quantum computers on the horizon [39, 48, 65].

Some secure archival works address this problem by leveraging unorthodox *information-theoretic* (also referred to as unconditional, or non-cryptographic) techniques that yield provably long-term security guarantees, but suffer higher storage costs than replication. Other works devise more storage-efficient cryptographic schemes; alas, due to their cryptographic nature, they still fail to provide long-term guarantees. Ultimately, the trade-off between efficiency and security is a central theme of secure archival, and we still have yet to see any system that strikes a desirable point in this trade-off design space.

In this work, we investigate the nature of this trade-off and related challenges in secure archival arising from the threat of cryptographic obsolescence. We study how these challenges are (or fail to be) addressed by existing solutions, and conclude with promising research directions.

2 Background and Motivation

To discuss secure archival issues, we first establish our security goals for archival systems. We then discuss how to formalize adversaries that attack the system, and introduce two notions of security that are central in what follows. Throughout the paper we operate under the standard assumption (for reasons of fault tolerance) that an archival system will, at a minimum, span geographically dispersed storage nodes [25, 27, 38, 43, 64]. We conclude the section with some related survey articles in the domain.

Security Goals. When discussing the goals of information security, we adopt the classic CIA triad of confidentiality, integrity, and availability. Confidentiality and integrity refer to the protection of data from unauthorized access and modification, respectively; availability refers to the ability of the system to provide correct functionality to its users upon request. These are not the only possible goals of information security (*e.g.*, non-repudiation), but the CIA triad is widely agreed upon, and most additional concerns can be lumped under the CIA umbrella. In this work we mostly focus on (long-term) confidentiality and integrity, since we consider availability to be much better understood in the storage community [25–27]. Availability can also be violated by purely physical means; there's no protection against a sufficiently determined adversary.

Threat Modeling. To violate any of the security goals, an adversary needs at least to corrupt some storage nodes. Obviously we cannot allow an adversary to corrupt all nodes, so typically an adversary is only allowed to corrupt at most a threshold number of nodes at any point in time. This idea is formalized in Ostrovsky and Yung's popular *mobile adversary model* [49], which is highly relevant for the provable security of many modern secure-archival works.

Adversarial models come in varying levels of formalism. Rigorous cryptographic adversarial models—like the mobile adversary model—allow one to obtain more useful security proofs, and are therefore considered the "gold standard" for adversarial modeling [21]. A key consideration in formulating an adversarial model is specifying its computational power. Typically, adversaries are viewed as Turing machines with either probabilistic polynomial runtime (PPT) or completely unbounded runtime, but some works make more nuanced computational assumptions. For instance, one can introduce real-time notions into the model and bound the rate of computation per unit of real time [17]. Additionally, one can define an adversary as a sequence of adversaries indexed by time, with each successive adversary belonging to a more powerful class of computing machines [15]. In this work we consider a mobile adversary with computational power bounded in this more nuanced manner; yet, for the purpose of understanding the challenges of secure archival it is often instructive to consider a computationally unbounded adversary, even though unbounded computing machines do not exist in the real world due to physical limits of computation [40].

Computational vs. Information-Theoretic Security. We draw an important distinction between *computational* and *information-theoretic* notions of security. If an adversarial model assumes any kind of restrictions on computing power, then we say that the security (if any) of our system with respect to some security goal is computational. Otherwise, it is information-theoretic.

Next, we present two security definitions for encryption as an example. Consider a key space $\mathcal{K}$, message space $\mathcal{M}$, and ciphertext space $\mathcal{C}$, and let $(\texttt{Enc},\ \texttt{Dec})$ be a (private key) encryption scheme such that $\texttt{Enc}\colon \mathcal{K}\times\mathcal{M}\to\mathcal{C}$, $\texttt{Dec}\colon \mathcal{K}\times\mathcal{C}\to\mathcal{M}$, and $\texttt{Dec}_k(\texttt{Enc}_k(m)) = m$ for all $k\in\mathcal{K},\ m\in\mathcal{M}$.

Definition 2.1 (ϵ-Statistically Indistinguishable Encryptions)**.** An encryption scheme $(\texttt{Enc},\texttt{Dec})$ is ϵ-statistically indistinguishable ($\epsilon > 0$) if, $\forall m_0, m_1 \in \mathcal{M}$, and for any $\mathcal{A}\colon \mathcal{C}\to\{0,1\}$:

$$|\Pr[\mathcal{A}(\texttt{Enc}_k(m_0)) = 1] - \Pr[\mathcal{A}(\texttt{Enc}_k(m_1)) = 1]| \le \epsilon$$

Definition 2.2 (ϵ-Computationally Indistinguishable Encryptions)**.** An encryption scheme $(\texttt{Enc},\texttt{Dec})$ is ϵ-computationally indistinguishable ($\epsilon > 0$) if, $\forall m_0, m_1 \in \mathcal{M}$, and for any PPT algorithm $\mathcal{A}\colon \mathcal{C}\to\{0,1\}$:

$$|\Pr[\mathcal{A}(\texttt{Enc}_k(m_0)) = 1] - \Pr[\mathcal{A}(\texttt{Enc}_k(m_1)) = 1]| \le \epsilon$$

Both definitions try to formalize the sentiment that an adversary $\mathcal{A}$ cannot distinguish between any two ciphertexts (*i.e.*, learn *any* information) with greater than a certain probability. The only difference is that in Definition 2.2, the adversary's computational power is restricted. Thus, this definition provides computational security, while Definition 2.1 provides the (much stronger) information-theoretic security. The computational setting is almost always preferred

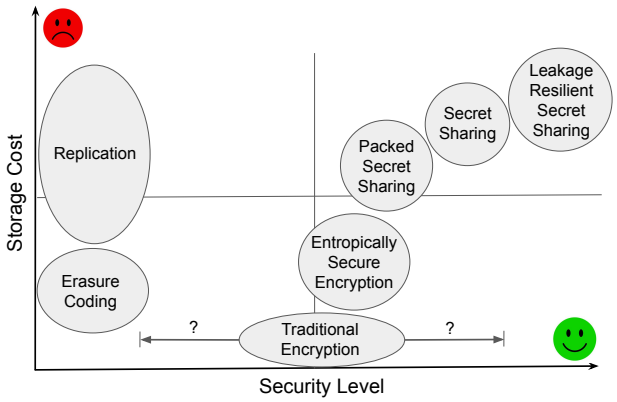

Figure 1. Qualitative quadrant graph of the storage cost vs. security trade-off for some data encodings. Traditional encryption refers to all computationally secure encryptions. Secure archival systems should come as close as possible to the smiley face.

over the information-theoretic one for efficiency reasons—especially storage cost (see Figure 1). However, information-theoretic definitions appear in many secure archival works due to the unique challenges posed by long-term security.

Related work. Several have studied the security of storage systems [13, 61, 66, 68], but only one [61] takes into account both confidentiality and integrity of long-lived data. For example, Braun *et al.* [13] study the confidentiality of storage systems and examine various information-theoretic key agreement solutions for data in transit and proactive secret sharing schemes for data at rest. Their work is complemented by the survey on integrity by Vigil *et al.* [66], which analyzes and compares solutions to integrity concerns including proof of existence, non-repudiation, and authenticity of documents. However, Braun *et al.* [13] and Vigil *et al.* [66] do not explore solutions that provide both confidentiality and integrity of long-term archival systems.

A survey by Yang *et al.* [68] investigates data security and privacy issues and potential countermeasures in cloud storage systems. The threats mentioned in their paper can be applicable to secure archives, but the focus is not long-term security. The survey by Storer *et al.* [61] highlights security threats specific to long-term archival systems. Our work can be thought of as an extension of Storer's in that we also consider some of the attacks mentioned in their paper and evaluate recent approaches to address them.

3 Challenges and Solutions

As outlined in Section 1, the primary security challenge for archival systems is the threat of cryptographic obsolescence. After briefly examining why this threat is (currently) unavoidable, we explore why developing a practical solution to overcome it is so difficult by stepping through several existing approaches, discussing confidentiality and integrity separately.

3.1 Cryptographic Obsolescence

The security of all computationally secure cryptographic primitives (encryption schemes, hash functions, etc.) relies on assumptions of hardness. For example, Diffie-Hellman key exchange assumes the hardness of computing discrete logarithms [20], RSA encryption assumes the hardness of factoring RSA moduli [54], several post-quantum cryptographic schemes assume the hardness of the Module Learning with Errors (MLWE) problem [11], and AES is simply assumed to be secure because two decades of cryptanalysis have failed to produce practical attacks [9]. All these assumptions rely on the existence of one-way functions [35]. Informally, a function is one-way if it is easy to compute, but any PPT algorithm fails to invert it with overwhelming probability. Unfortunately, the existence of one-way functions directly implies $\mathbb{P} \neq \mathbb{NP}$ [5]. Consequently, any computationally secure cryptographic primitive can potentially be broken in the future. Of course, this has not prevented security practitioners from developing schemes that are widely believed to be secure. Yet, history has repeatedly shown that schemes believed to be secure at one point in time are often broken in the future, like the MD5 hash [60], DES encryption [8], and discrete-log based schemes against quantum computers [58]. Thus, trying to extrapolate the computational security of current cryptographic schemes decades or more into the future is a fairly risky endeavor.

3.2 Confidentiality

To simplify exposition of existing approaches, we begin by addressing data confidentiality, grouping approaches by computational vs. information-theoretic notions of security.

Computational security. It is natural to ask whether we can still use our arsenal of computationally secure cryptography to achieve long-term confidentiality in the face of cryptographic obsolescence. For instance, can we not just periodically re-encrypt data? A naïve approach is for the user to retrieve data, decrypt, and re-encrypt it with a new scheme, but this requires user intervention. This is especially problematic in an emergency situation where the current encryption method is broken, and *all* users' data needs to be re-encrypted immediately. This re-encryption could be delegated to the storage system (without giving the system access to user keys) using more sophisticated techniques like Universal Proxy Re-Encryption (UPRE) [23].

Unfortunately, regardless of technique, it may be infeasible to re-encrypt all data in a timely manner due to I/O bottlenecks. An archival system will require massive amounts of storage media that is cost effective and easy to secure. Therefore, we exclude HDDs and SSDs as they are (i) too expensive and (ii) less secure, because online media are more prone to

remote attacks. Conversely, removable media stored offline enjoy a reduced attack surface (*e.g.*, DVDs, tapes). We illustrate with a few examples of tape-based archives—a common archival storage medium [16]—in the literature. A similar exercise can be conducted with glass-based media [4].

In this example, we attempt to use conservative numbers from the cited works below—whether explicit or implicit—and make several simplifying assumptions: a real archive would have even longer re-encryption times and costs. A conservative approximation for the time to just *read* all the data in an archive can be obtained by dividing the size of the archive by its aggregate read throughput. With 80PB of total data and an aggregate throughput of 400TB/day, the Oak Ridge HPSS [59] could be read in 6.75 months. At 37.9PB and 120TB/day, the ECMWF MARS archive [30] yields 10.35 months, and the CERN EOS archive [51] yields 8.3 months for 230PB and 909TB/day. Pergamum [62] (a HDD archive) describes a hypothetical 10PB tape archive with aggregate throughput of 5GB/s, yielding 0.76 months. In addition to reading, the system has to encrypt (*i.e.*, consume memory and CPU) and then write out the data. Writing in many systems tends to be slower than reading due to media limitations and the need to verify written data; this factor will at least double the re-encryption duration. Moreover, in practice the system would *have* to reserve some capacity for new activity (*e.g.*, ingest new data and serve read requests); this additional factor can easily double the re-encryption duration. Lastly, the aforementioned systems are considerably smaller than the archives we envision in the many exabyte and even zettabyte sizes over their lifetimes. All things considered, the practical time for re-encrypting an entire archive could turn into many years—during which time all not-yet-encrypted data remains vulnerable.

One could avoid the I/O cost of re-encryption—at the cost of storing a growing history of encryption keys—by using *multiple* layers of different encryption schemes to hedge against the threat of any one or more ciphers being broken. This approach, known as a *robust combiner*, more precisely a *cascade cipher*, is used in the secure-archival system ArchiveSafeLT [56]. Cascade ciphers enjoy the property of being at least as secure as the most secure cipher in the cascade [33], but care must be taken in their design as failure to account for subtle considerations can render the whole cascade insecure [45]. ArchiveSafeLT also proposes wrapping data in new layers of encryption if enough of the old layers are broken, though this runs into the same I/O issues as re-encryption.

Another approach is that taken in AONT-RS [53], which was deployed as part of the Cleversafe dispersed storage system (later acquired by IBM's Cloud Object Storage). Let $\texttt{Enc}_k(\cdot)$ denote a computationally secure encryption using key k, and $h(\cdot)$ a computationally secure hash function. The AONT-RS scheme begins by splitting the data to be encrypted into equal-sized blocks $m_1, ..., m_s$. Then, for each $i \in \{1, ..., s\}$ the scheme computes ciphertext blocks $c_i = m_i \oplus \texttt{Enc}_k(i+1)$, and a final ciphertext block $c_{s+1} = k \oplus h(c_1, ..., c_s)$. The $s+1$ ciphertext blocks are encoded into $n > s+1$ codewords via systematic erasure coding, and each codeword is then dispersed to a different storage node. Assuming Enc and h are computationally secure, then a PPT attacker provably cannot learn any information about the plaintext unless they possess all s+1 ciphertext blocks or know the key. This scheme is easy to parallelize, achieves a good trade-off between storage cost and availability, and eliminates the need for key management. On the other hand, if the underlying encryption scheme or hash function are broken, an attacker trivially "knows the key" and can recover plaintext from even a single share. Note that, apart from AONT-RS, every other commercially available archival system we are aware of simply uses AES (*e.g.*, AWS, Google Cloud, Azure [1–3]).

We have seen how to devise computationally secure methods that improve upon naïve re-encryption. However, these methods are *all* still vulnerable to a showstopping attack: they are susceptible to *Harvest Now, Decrypt Later* attacks. Re-encryption does nothing to protect portions of any stolen ciphertext.

Information-theoretic security. The simplest example of information-theoretically secure encryption is the One-Time Pad [36]. To encrypt a message m, a key $k \in \{0,1\}^{|m|}$ is sampled uniformly at random, and the ciphertext is computed as $c = m \oplus k$. Without knowledge of k, an adversary provably cannot learn any information about m from c regardless of computing power, thus achieving "perfect secrecy" (*i.e.*, let $\epsilon = 0$ in Definition 2.1).

A generalization of the One-Time Pad is Shamir's secret sharing [57]. It takes a message m as input, and outputs n shares $s_1, ..., s_n$, with $|s_i| = |m|$, such that any subset of $t \leq n$ or more shares suffices to recover m, but fewer than t shares leaves m perfectly secret. The mechanism is equivalent [46] to a non-systematic $[n, t]$ Reed-Solomon code applied to $(m, r_1, ..., r_{t-1})$, where m is the message and $r_1, ..., r_{t-1}$ are all sampled uniformly at random from $\{0,1\}^{|m|}$.

Many secure archival systems propose using secret sharing as a data encoding scheme, thus providing long-term confidentiality [12, 14, 27, 28, 47, 63, 64, 67]. POTSHARDS [63] was the first work to design and evaluate a full archival system based on Shamir's secret sharing. In POTSHARDS, each share is uploaded to an administratively independent storage provider, thereby avoiding a single point of trust or failure, and achieving good availability due to the erasure-coding properties of Shamir's secret sharing and some additional disaster recovery techniques. There are two main drawbacks with this scheme. The first is high storage overhead: each share is the same size as the original message, so we incur the same overhead as replication with less availability as we can tolerate the loss of at most $n - t$ shares. This cost is a provably unavoidable consequence of perfect secrecy [6].

4

The second is that given enough time, we must entertain the possibility that a mobile adversary eventually steals a threshold number of shares.

To combat this mobile adversary, it is desirable for the system to have a means of "refreshing" the shares, rendering stolen shares obsolete. This can be accomplished via proactive secret sharing [34]: an information-theoretic distributed protocol that re-randomizes shares. Wong *et al.* [67] suggest using a version of proactive secret sharing for secure archival with the desirable feature of adding or removing shareholders in each share renewal phase. Unfortunately, share renewal requires every shareholder to send a share to each shareholder. This incurs high communication costs. Moreover, if the system must execute share renewal for many data objects in a short time frame, this may become impractical for the same reasons as re-encryption. Nonetheless, proactive secret-shared datastores remain the leading (and only) approach for secure-archival systems that provide long-term information-theoretic confidentiality of data at rest.

Given the strong information-theoretic security of secret-shared datastores, an adversary may find it more fruitful to steal data in transit rather than data at rest, since TLS encryption is only computationally secure. This motivates a desire for information-theoretically secure communication channels. LINCOS [12] and follow-up works [14, 28, 47] construct information-theoretic channels via Quantum Key Distribution (QKD). By setting up entangled quantum states, two parties can generate a shared One-Time Pad key that is impervious to eavesdropping. While promising, QKD requires specialized infrastructure, and a number of engineering challenges must be resolved before QKD can be considered a mature, practical, and cost-effective method [18]. Another approach to information-theoretic channels is introduced in Section 4.

3.3 Integrity

Whereas long-term confidentiality guarantees require expensive information-theoretic methods, long-term integrity is more amenable to cryptographic tools. Today, computationally secure digital signatures are widely used for integrity protection. A single signature alone may eventually be broken, but long-term integrity can be achieved with a chain of digitally signed timestamps [32]. We omit details, but intuitively this works because signing an old signature with a new signature preserves the integrity of both as long as the old signature has not been broken at the time the new signature was computed. Note that a computationally unbounded attacker could instantly break this signature with brute force, but unbounded attackers are unnecessarily strict in this scenario. A signature's integrity needs to hold for only a relatively short interval of real time until a newer, more secure signature is added to the chain. This motivates the more nuanced computationally bounded adversary described in Section 2. LINCOS makes the key observation that while timestamp chains provide long-term integrity, the use of computationally secure hashes within the chain compromises the information-theoretic confidentiality of data. They remedy this by swapping out hashes with information-theoretically hiding commitments, such as the Pedersen commitment [50].

Timestamp chains provide long-term integrity of a single data object in isolation, but this by itself is insufficient for secret-shared datastores, where one must also ensure that shares are consistent with each other. This is especially important for the share renewal phase of proactive secret sharing, as a corrupt shareholder that distributes invalid new shares can compromise the integrity of the secret. Verifiable secret sharing [50] protects against this threat, and is often included by default as a sub-protocol of proactive secret sharing. The use of Pedersen commitments within verifiable secret sharing protocols is again useful in order to safeguard long-term confidentiality.

4 Discussion and Future Directions

In Section 3 we examined how different approaches to secure archival systems fare against the core challenge of cryptographic obsolescence. We found that computationally secure methods can yield decent long-term integrity, and are sometimes (*e.g.*, AONT-RS) practical enough to make their way into commercial use, but fail to provide long-term confidentiality guarantees. Conversely, the information-theoretic approach of a secret-shared datastore equipped with proactive secret sharing and information-theoretic channels (via QKD) delivers strong long-term confidentiality, but suffers from high storage and communication overheads, as well as higher infrastructure costs.

We are left with a seemingly intractable trade-off between efficiency and security. A reasonable response to this state of affairs is to declare that there is no "one size fits all" approach to secure archival. This statement was made nearly two decades ago by the PASIS [27] project, which investigated several approaches but left users to decide which one was best for their data. This may very well be the case, but the unsavory trade-off remains. See Table 1 for a summary of systems discussed in this work.

We conclude with some future directions for alleviating the trade-off by reducing costs with alternative storage media, and improving the security of information-theoretic systems with new and existing techniques.

The high storage costs of secret-shared datastores may be reduced with cheaper and denser archival storage media. One leading candidate is DNA storage, with a theoretical density of 1EB per cubic millimeter (8 orders of magnitude greater than tape), and centuries of durability [10]. Citing the high costs and low throughputs of DNA synthesis and sequencing, Microsoft's Project Silica [4] advocates for glass as an archival storage medium. While not as dense as DNA (only 429TB per cubic inch [69]), glass requires very little

Systems	Confidentiality		Storage Cost
	In Transit	At Rest	
ArchiveSafeLT [56]	Computational	Computational	Low
AONT-RS [53]	Computational	Computational	Low
HasDPSS [70]	Computational	ITS	High
LINCOS [12]	ITS	ITS	High
PASIS [27]	Computational	ITS (sometimes)	Low-High
POTSHARDS [63]	Computational	ITS	High
VSR Archive [67]	Computational	ITS	High
AWS, Azure, Google Cloud [1–3]	Computational	Computational	Low

Table 1. Summary of discussed systems. ITS stands for information-theoretic security.

maintenance, can survive for millenia, and is much closer to widespread adoption than DNA storage. Photosensitive film is also low maintenance, potentially lasts centuries, and is currently used in the the Arctic World Archive [55].

Instead of stealing an entire secret share from the archive, an adversary might leak only a few bits of information about a share via some hidden side-channel. Shamir's secret sharing is known to be vulnerable to such leakage attacks [7]; several recent works [7, 19, 37, 41, 42] have proposed new leakage-resilient secret sharing (LRSS) schemes. Evaluating LRSS's viability for archival systems is an open problem. In particular, LRSS schemes can be broadly classified according to leakage model and linearity. The leakage model determines the class of side-channel attacks that the scheme can resist; it is unclear which leakage model is most appropriate for a secret-shared archive. Linear LRSS constructions tend to be simple and compatible with existing proactive secret-sharing schemes, but their security often imposes restrictions on what the threshold value can be. Nonlinear schemes have no such restrictions, but their constructions are more complex, and it is unclear how to make these schemes proactive.

An alternative to QKD for information-theoretic channels is the Bounded Storage Model (BSM) [44]. In the BSM, honest parties can agree on a One-Time Pad key by streaming large amounts of random data to each other such that an adversary with a much larger storage capacity cannot capture the entire stream. We believe the BSM is overdue for a practical evaluation—last evaluated in 2005 [24]. Since then, new theoretical results have expanded the possibilities of the BSM. Namely, the necessary gap between honest and adversarial storage has been improved from quadratic to exponential for important cryptographic primitives like key agreement, oblivious transfer, and general multi-party computation, at the cost of increased round and communication complexity [22, 31, 52]. It remains to be seen whether these costs are low enough in practice to consider the BSM as a viable option for information-theoretic channels.

The concrete design and implementation of secret-shared archives may benefit from the literature on key-management systems, as their architectures can be quite similar. For example, HasDPSS [70] leverages modern blockchain and proactive secret-sharing techniques to realize a robust and decentralized key-management system.

To conclude, the currently unavoidable threats of cryptographic obsolescence and Harvest Now, Decrypt Later attacks induce a steep trade-off between security and storage cost that depends on whether designers of secure archives choose between computational or information-theoretically secure techniques. Fundamentally, this trade-off will remain unless large open problems in complexity theory are resolved (see Section 3.1), but exploring certain research directions such as next-generation archival media, LRSS schemes, BSM improvements, and recent developments in key-management systems may make the trade-off more palatable by improving information-theoretic systems.

References

[1] 2023. *Azure Storage encryption for data at rest.* https://learn.microsoft.com/en-us/azure/storage/common/infrastructure-encryption-enable?tabs=portal

[2] 2024. *Amazon S3 now automatically encrypts all new objects.* https://docs.aws.amazon.com/AmazonS3/latest/userguide/default-encryption-faq.html

[3] 2024. *Default encryption at rest | documentation | google cloud.* https://cloud.google.com/docs/security/encryption/default-encryption

[4] Patrick Anderson, Erika B. Aranas, Youssef Assaf, Raphael Behrendt, Richard Black, Marco Caballero, Pashmina Cameron, Burcu Canakci, Thales de Carvalho, Andromachi Chatzieleftheriou, James Clegg, Rebekah Storan Clarke, Daniel Cletheroe, Bridgette Cooper, Tim Deegan, Austin Donnelly, Rokas Drevinskas, Alexander Gaunt, Christos Gkantsidis, Ariel Gomez Diaz, Istvan Haller, Freddie Hong, Teodora Ilieva, Shashidhar Joshi, Russell Joyce, Mint Kunkel, David Lara, Sergey Legtchenko, Fanglin Linda Liu, Bruno Magalhaes, Alana Marzoev, Marvin McNett, Jayashree Mohan, Michael Myrah, Truong Nguyen, Sebastian Nowozin, Aaron Ogus, Hiske Overweg, Ant Rowstron, Maneesh Sah, Masaaki Sakakura, Peter Scholtz, Nina Schreiner, Omer Sella, Adam Smith, Ioan Stefanovici, David Sweeney, Benn Thomsen, Govert Verkes, Phil Wainman, Jonathan Westcott,

Luke Weston, Charles Whittaker, Pablo Wilke Berenguer, Hugh Williams, Thomas Winkler, and Stefan Winzeck. 2023. Project Silica: Towards Sustainable Cloud Archival Storage in Glass. In *The 29th ACM Symposium on Operating Systems Principles*. https://www.microsoft.com/en-us/research/publication/project-silica-towards-sustainable-cloud-archival-storage-in-glass/

[5] Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity: A Modern Approach* (1st ed.). Cambridge University Press, USA.

[6] Amos Beimel. 2011. Secret-Sharing Schemes: A Survey. In *Coding and Cryptology*, Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 11–46.

[7] Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. 2021. On the local leakage resilience of linear secret sharing schemes. *Journal of Cryptology* 34 (2021), 1–65. https://link.springer.com/article/10.1007/s00145-021-09375-2.

[8] Eli Biham and Adi Shamir. 1993. *Differential Cryptanalysis of the Data Encryption Standard*. Springer. https://doi.org/10.1007/978-1-4613-9314-6

[9] Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. 2011. Biclique Cryptanalysis of the Full AES. In *Advances in Cryptology – ASIACRYPT 2011*, Dong Hoon Lee and Xiaoyun Wang (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 344–371.

[10] James Bornholt, Randolph Lopez, Douglas M. Carmean, Luis Ceze, Georg Seelig, and Karin Strauss. 2017. Toward a DNA-Based Archival Storage System. *IEEE Micro* 37, 3 (2017), 98–104. https://doi.org/10.1109/MM.2017.70

[11] Joppe Bos, Leo Ducas, Eike Kiltz, T Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehle. 2018. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. 353–367. https://doi.org/10.1109/EuroSP.2018.00032

[12] Johannes Braun, Johannes Buchmann, Denise Demirel, Matthias Geihs, Mikio Fujiwara, Shiho Moriai, Masahide Sasaki, and Atsushi Waseda. 2017. LINCOS: A Storage System Providing Long-Term Integrity, Authenticity, and Confidentiality. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (Abu Dhabi, United Arab Emirates) *(ASIA CCS '17)*. Association for Computing Machinery, New York, NY, USA, 461–468. https://doi.org/10.1145/3052973.3053043 https://dl.acm.org/doi/10.1145/3052973.3053043.

[13] Johannes Braun, Johannes Buchmann, Ciaran Mullan, and Alex Wiesmaier. 2014. Long term confidentiality: a survey. *Designs, Codes and Cryptography* 71 (2014), 459–478. https://eprint.iacr.org/2012/449.pdf.

[14] Johannes Buchmann, Ghada Dessouky, Tommaso Frassetto, Ágnes Kiss, Ahmad-Reza Sadeghi, Thomas Schneider, Giulia Traverso, and Shaza Zeitouni. 2020. SAFE: A Secure and Efficient Long-Term Distributed Storage System. In *Proceedings of the 8th International Workshop on Security in Blockchain and Cloud Computing* (Taipei, Taiwan) *(SBC '20)*. Association for Computing Machinery, New York, NY, USA, 8–13. https://doi.org/10.1145/3384942.3406868

[15] Ahto Buldas, Matthias Geihs, and Johannes Buchmann. 2017. Long-Term Secure Time-Stamping using Preimage-Aware Hash Functions. *IACR Cryptol. ePrint Arch.* (2017), 754. http://eprint.iacr.org/2017/754

[16] James Byron. 2022. *Modeling the Future of Archival Storage Systems*. Ph.D. Dissertation.

[17] Ran Canetti, Ling Cheung, Dilsun Kaynar, Nancy Lynch, and Olivier Pereira. 2008. Modeling Computational Security in Long-Lived Systems. In *CONCUR 2008 - Concurrency Theory*, Franck van Breugel and Marsha Chechik (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 114–130.

[18] Yuan Cao, Yongli Zhao, Qin Wang, Jie Zhang, Soon Xin Ng, and Lajos Hanzo. 2022. The evolution of quantum key distribution networks: On the road to the qinternet. *IEEE Communications Surveys & Tutorials* 24, 2 (2022), 839–894.

[19] Nishanth Chandran, Bhavana Kanukurthi, Sai Lakshmi Bhavana Obbattu, and Sruthi Sekar. 2022. Short Leakage Resilient and Non-malleable Secret Sharing Schemes. In *Annual International Cryptology Conference*. Springer, 178–207. https://eprint.iacr.org/2022/216.pdf.

[20] W. Diffie and M. Hellman. 1976. New directions in cryptography. *IEEE Transactions on Information Theory* 22, 6 (1976), 644–654. https://doi.org/10.1109/TIT.1976.1055638

[21] Quang Do, Ben Martini, and Kim-Kwang Raymond Choo. 2019. The role of the adversary model in applied security research. *Computers & Security* 81 (2019), 156–181. https://doi.org/10.1016/j.cose.2018.12.002

[22] Yevgeniy Dodis, Willy Quach, and Daniel Wichs. 2023. Speak Much, Remember Little: Cryptography in the Bounded Storage Model, Revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 86–116.

[23] Nico Döttling and Ryo Nishimaki. 2021. Universal Proxy Re-Encryption. In *Public-Key Cryptography - PKC 2021 - 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10-13, 2021, Proceedings, Part I (Lecture Notes in Computer Science)*, Juan A. Garay (Ed.), Vol. 12710. Springer, 512–542. https://doi.org/10.1007/978-3-030-75245-3_19

[24] Timothy John Draelos, William Douglas Neumann, Andrew J Lanzone, and William Erik Anderson. 2005. *Key management and encryption under the bounded storage model*. Technical Report. Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA ....

[25] Seth Eliot, Mahanth Jayadeva, Amulya Sharma, Jason DiDomenico, Marcin Bednarz, Tyler Applebaum, Rodney Lester, Joe Chapman, Adrian Hornsby, Kevin Miller, Shannons Richards, Laurent Domb, Kevin Schwarz, Rob Martell, Priyam Reddy, Jeff Ferris, and Matias Battaglia. 2024. *Reliability Pillar: AWS Well-Architected Framework*. White Paper. Amazon Web Services. https://docs.aws.amazon.com/wellarchitected/latest/reliability-pillar/welcome.html.

[26] Daniel Ford, François Labelle, Florentina I Popovici, Murray Stokely, Van-Anh Truong, Luiz Barroso, Carrie Grimes, and Sean Quinlan. 2010. Availability in globally distributed storage systems. In *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10)*.

[27] Gregory R Ganger, Pradeep K Khosla, and CARNEGIE-MELLON UNIV PITTSBURGH PA. 2005. PASIS: A Distributed Framework for Perpetually Available and Secure Information Systems. https://apps.dtic.mil/sti/citations/ADA436245.

[28] Matthias Geihs, Nikolaos Karvelas, Stefan Katzenbeisser, and Johannes Buchmann. 2018. PROPYLA: Privacy Preserving Long-Term Secure Storage *(SCC '18)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/3201595.3201599

[29] Phil Goodwin. 2019. *Tape and Cloud: Solving Storage Problems in the Zettabyte Era of Data*. White Paper. International Data Corporation (IDC). https://www.lto.org/wp-content/uploads/2019/06/Tape-and-Cloud-Solving-Storage-Problems-in-the-Zettabyte-Era.pdf

[30] Matthias Grawinkel, Lars Nagel, Markus Mäsker, Federico Padua, André Brinkmann, and Lennart Sorth. 2015. Analysis of the ECMWF storage landscape. In *Proceedings of the 13th USENIX Conference on File and Storage Technologies* (Santa Clara, CA) *(FAST'15)*. USENIX Association, USA, 15–27.

[31] Jiaxin Guan and Mark Zhandary. 2019. Simple schemes in the bounded storage model. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38*. Springer, 500–524.

[32] Stuart Haber and W Scott Stornetta. 1991. *How to time-stamp a digital document*. Springer.

[33] Amir Herzberg. 2009. Folklore, practice and theory of robust combiners. *Journal of Computer Security* 17, 2 (2009), 159–189.

[34] Amir Herzberg, Stanisław Jarecki, Hugo Krawczyk, and Moti Yung. 1995. Proactive secret sharing or: How to cope with perpetual leakage. In *Advances in Cryptology—CRYPT0'95: 15th Annual International Cryptology Conference Santa Barbara, California, USA, August 27–31, 1995 Proceedings 15*. Springer, 339–352. https://link.springer.com/chapter/10.1007/3-540-44750-4_27.

[35] R. Impagliazzo. 1995. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*. 134–147. https://doi.org/10.1109/SCT.1995.514853

[36] Jonathan Katz and Yehuda Lindell. 2014. *Introduction to Modern Cryptography, Second Edition*. CRC Press. https://www.crcpress.com/Introduction-to-Modern-Cryptography-Second-Edition/Katz-Lindell/p/book/9781466570269

[37] Ohad Klein and Ilan Komargodski. 2023. New Bounds on the Local Leakage Resilience of Shamir's Secret Sharing Scheme. In *Advances in Cryptology – CRYPTO 2023*, Helena Handschuh and Anna Lysyanskaya (Eds.). Springer Nature Switzerland, Cham, 139–170. https://link.springer.com/chapter/10.1007/978-3-031-38557-5_5.

[38] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishan Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Zhao. 2000. OceanStore: an architecture for global-scale persistent storage. *SIGPLAN Not.* 35, 11 (nov 2000), 190–201. https://doi.org/10.1145/356989.357007

[39] David Lague. 2023. U.S. and China race to shield secrets from quantum computers. *Reuters* (2023). https://www.reuters.com/investigates/special-report/us-china-tech-quantum

[40] R. Landauer. 1961. Irreversibility and Heat Generation in the Computing Process. *IBM Journal* (July 1961).

[41] Hemanta K. Maji, Hai H. Nguyen, Anat Paskin-Cherniavsky, Tom Suad, Mingyuan Wang, Xiuyu Ye, and Albert Yu. 2022. Leakage-resilient Linear Secret-sharing Against Arbitrary Bounded-size Leakage Family. In *Theory of Cryptography*, Eike Kiltz and Vinod Vaikuntanathan (Eds.). Springer Nature Switzerland, Cham, 355–383. https://link.springer.com/chapter/10.1007/978-3-031-22318-1_13.

[42] Hemanta K Maji, Anat Paskin-Cherniavsky, Tom Suad, and Mingyuan Wang. 2021. Constructing locally leakage-resilient linear secret-sharing schemes. In *Annual International Cryptology Conference*. Springer, 779–808.

[43] Petros Maniatis, Mema Roussopoulos, T. J. Giuli, David S. H. Rosenthal, and Mary Baker. 2005. The LOCKSS peer-to-peer digital preservation system. *ACM Trans. Comput. Syst.* 23, 1 (feb 2005), 2–50. https://doi.org/10.1145/1047915.1047917

[44] Ueli M Maurer. 1992. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology* 5 (1992), 53–66.

[45] Ueli M Maurer and James L Massey. 1993. Cascade ciphers: The importance of being first. *Journal of Cryptology* 6 (1993), 55–61.

[46] R. J. McEliece and D. V. Sarwate. 1981. On sharing secrets and Reed-Solomon codes. *Commun. ACM* 24, 9 (sep 1981), 583–584. https://doi.org/10.1145/358746.358762

[47] Philipp Muth, Matthias Geihs, Tolga Arul, Johannes Buchmann, and Stefan Katzenbeisser. 2020. ELSA: efficient long-term secure storage of large datasets (full version). *EURASIP Journal on Information Security* 2020 (2020), 1–20.

[48] Greg Noone. 2023. Are harvest now, decrypt later cyberattacks actually happening? *Tech Monitor* (2023). https://techmonitor.ai/hardware/quantum/harvest-now-decrypt-later-cyberattack-quantum-computer

[49] Rafail Ostrovsky and Moti Yung. 1991. How to Withstand Mobile Virus Attacks (Extended Abstract). In *Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing* (Montreal, Quebec, Canada) *(PODC '91)*. Association for Computing Machinery, New York, NY, USA, 51–59. https://doi.org/10.1145/112600.112605 https://doi.org/10.1145/112600.112605.

[50] Torben Pryds Pedersen. 1991. Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual international cryptology conference*. Springer, 129–140.

[51] Devashish R. Purandare, Daniel Bittman, and Ethan L. Miller. 2022. Analysis and workload characterization of the CERN EOS storage system. In *Proceedings of the Workshop on Challenges and Opportunities of Efficient and Performant Storage Systems* (Rennes, France) *(CHEOPS '22)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3503646.3524293

[52] Ran Raz. 2018. Fast learning requires good memory: A time-space lower bound for parity learning. *Journal of the ACM (JACM)* 66, 1 (2018), 1–18.

[53] Jason K. Resch and James S. Plank. 2011. AONT-RS: blending security and performance in dispersed storage systems. In *Proceedings of the 9th USENIX Conference on File and Stroage Technologies* (San Jose, California) *(FAST'11)*. USENIX Association, USA, 14.

[54] R. L. Rivest, A. Shamir, and L. Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* 21, 2 (feb 1978), 120–126. https://doi.org/10.1145/359340.359342

[55] Jędrzej Sabliński and Alfredo Trujillo. 2021. Piql. Long-term preservation technology study. *Archeion* 2021, 122 (2021). https://www.ejournals.eu/Archeion/2021/122/art/20806/

[56] Moe Sabry and Reza Samavi. 2022. ArchiveSafe LT: Secure Long-term Archiving System *(ACSAC '22)*. Association for Computing Machinery, New York, NY, USA, 936–948. https://doi.org/10.1145/3564625.3564635

[57] Adi Shamir. 1979. How to Share a Secret. *Commun. ACM* 22, 11 (nov 1979), 612–613. https://doi.org/10.1145/359168.359176 https://doi.org/10.1145/359168.359176.

[58] Peter W. Shor. 1999. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Rev.* 41, 2 (1999), 303–332. https://doi.org/10.1137/S0036144598347011

[59] Hyogi Sim and Sudharshan S. Vazhkudai. 2019. Profiling the Usage of an Extreme-Scale Archival Storage System. In *2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. 410–422. https://doi.org/10.1109/MASCOTS.2019.00050

[60] Marc Stevens, Alexander Sotirov, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, and Benne de Weger. 2009. Short Chosen-Prefix Collisions for MD5 and the Creation of a Rogue CA Certificate. In *Advances in Cryptology - CRYPTO 2009*, Shai Halevi (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 55–69.

[61] Mark W. Storer, Kevin M. Greenan, and Ethan L. Miller. 2006. Long-term threats to secure archives. In *Proceedings of the 2006 ACM Workshop On Storage Security And Survivability, StorageSS 2006, Alexandria, VA, USA, October 30, 2006*, Ethan L. Miller and Erez Zadok (Eds.). ACM, 9–16. https://doi.org/10.1145/1179559.1179562

[62] Mark W. Storer, Kevin M. Greenan, Ethan L. Miller, and Kaladhar Voruganti. 2008. Pergamum: replacing tape with energy efficient, reliable, disk-based archival storage. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies* (San Jose, California) *(FAST'08)*. USENIX Association, USA, Article 1, 16 pages.

[63] Mark W. Storer, Kevin M. Greenan, Ethan L. Miller, and Kaladhar Voruganti. 2009. POTSHARDS—a Secure, Recoverable, Long-Term Archival Storage System. *ACM Trans. Storage* 5, 2, Article 5 (jun 2009), 35 pages. https://doi.org/10.1145/1534912.1534914 https://dl.acm.org/doi/10.1145/1534912.1534914.

[64] Arun Subbiah and Douglas M. Blough. 2005. An approach for fault tolerant and secure data storage in collaborative work environments. In

Proceedings of the 2005 ACM Workshop on Storage Security and Survivability (Fairfax, VA, USA) *(StorageSS '05)*. Association for Computing Machinery, New York, NY, USA, 84–93. https://doi.org/10.1145/1103780.1103793

[65] Kevin Townsend. 2022. Solving the Quantum Decryption 'Harvest Now, Decrypt Later' Problem. *SecurityWeek* (2022). https://www.securityweek.com/solving-quantum-decryption-harvest-now-decrypt-later-problem/

[66] Martín Vigil, Johannes Buchmann, Daniel Cabarcas, Christian Weinert, and Alexander Wiesmaier. 2015. Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: A survey. *Computers & Security* 50 (2015), 16–32. https://doi.org/10.1016/j.cose.2014.12.004

[67] T.M. Wong, Chenxi Wang, and J.M. Wing. 2002. Verifiable secret redistribution for archive systems. In *First International IEEE Security in Storage Workshop, 2002. Proceedings.* 94–105. https://doi.org/10.1109/SISW.2002.1183515 https://ieeexplore.ieee.org/document/1183515.

[68] Pan Yang, Naixue Xiong, and Jingli Ren. 2020. Data Security and Privacy Protection for Cloud Storage: A Survey. *IEEE Access* 8 (2020), 131723–131740. https://doi.org/10.1109/ACCESS.2020.3009876

[69] Jingyu Zhang, A Čerkauskaitė, Rokas Drevinskas, Aabid Patel, Martynas Beresna, and Peter G Kazansky. 2016. Eternal 5D data storage by ultrafast laser writing in glass. In *Laser-based Micro-and Nanoprocessing X*, Vol. 9736. Spie, 163–178.

[70] Yifang Zhang, Mingyue Wang, Yu Guo, and Fangda Guo. 2023. Towards Dynamic and Reliable Private Key Management for Hierarchical Access Structure in Decentralized Storage. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (<conf-loc>, <city>Birmingham</city>, <country>United Kingdom</country>, </conf-loc>) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 3371–3380. https://doi.org/10.1145/3583780.3615090