A coding theory approach to unconditionally secure proof-of-retrievability schemes for cloud storage

Douglas R. Stinson

David R. Cheriton School of Computer Science
University of Waterloo

5TH INTERNATIONAL SYMPOSIUM ON FOUNDATIONS & PRACTICE OF SECURITY

October 25–26, 2012
École de Technologie Supérieure, Montréal

This is joint work with Maura Paterson and Jalaj Upadhyay.

The problem setting

- *Alice* asks a *server* to store a (possibly large) file (or message) m (e.g., using cloud storage).
- The message m is divided into message blocks that we view as elements of a finite field.
- Typically, the message m will be encoded as M, using a public error-correcting code such as a Reed-Solomon code.
- The code provides redundancy, enabling erasures or corrupted message blocks in M to be corrected.
- Main problem: How can *Alice* be convinced that the *server* is storing the encoded message M correctly?
- Typical solution: A challenge-response protocol is periodically invoked by *Alice*.

Bounded-use schemes

- We do not assume that *Alice* is storing m or M.
- *Alice* must precompute and store a fixed number of challenge-response pairs, before transmitting M to the *server*.
- *Alice* gains confidence in the *server* if it is able to respond to all (or most of) her challenges.
- A *server* who can respond correctly to a large proportion of challenges should "know" (or be able to compute) the contents of the unencoded message m (i.e., all the message blocks).
- This idea is formalised in the notion of an extractor, in which case we have a proof-of-retrievability (or POR) scheme.

Extractors

- The *Extractor* takes as input a description of the *server*'s proving algorithm, denoted $\mathcal{P}$, and then outputs an unencoded message $\widehat{m}$.
- Extraction succeeds if $\widehat{m} = m$.
- The success probability of $\mathcal{P}$, denoted $\mathrm{succ}(\mathcal{P})$, is the probability that $\mathcal{P}$ gives a correct response for a randomly chosen challenge.
- Definition: the POR scheme is (δ, ϵ)-secure if the *Extractor* succeeds with probability at least δ whenever $\mathrm{succ}(\mathcal{P}) \geq \epsilon$.

Some previous related work

- Blum *et al.* (1994) introduced memory checking.
- Lillibridge *et al.* (2005) studied internet backup schemes.
- Naor and Rothblum (2005) studied online memory checkers and authenticators and they gave a lower bound on storage requirements and communication complexity.
- Juels and Kaliski (2007) introduced proof of retrievability schemes.
- Atieniese *et al.* (2007) introduced proof of data possession schemes.
- Shacham and Waters (2008) gave examples of unbounded-use schemes along with formal security proofs.
- Bowers, Juels, and Oprea (2009) used inner and outer codes to construct POR schemes.
- Dodis, Vadhan and Wichs (2009) gave the first examples of unconditionally secure POR schemes.

Three phases in a POR scheme

1. initialisation

Alice		*server*
$M = e(m)$	$\xrightarrow{M}$	*server* constructs $\mathcal{P}$

Three phases in a POR scheme

1. initialisation

Alice		*server*
$M = e(m)$	$\xrightarrow{M}$	*server* constructs $\mathcal{P}$

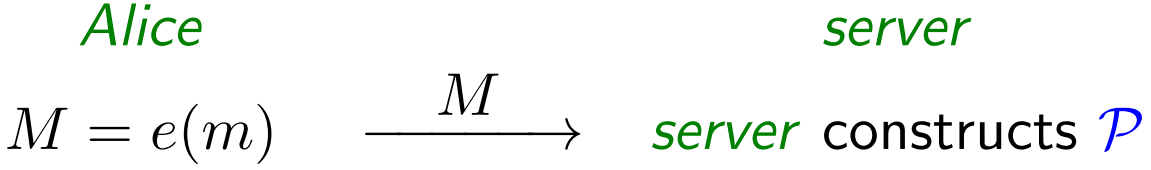
2. audit

Alice		*server*
	$\xrightarrow{c_i}$	
		$r_i = \mathcal{P}(c_i)$
	$\xleftarrow{r_i}$	

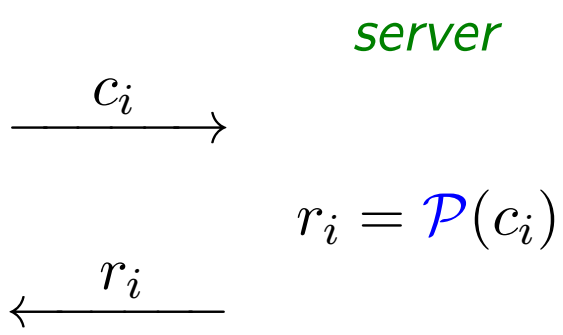
Here $i = 1, 2, \ldots$.

Three phases in a POR scheme

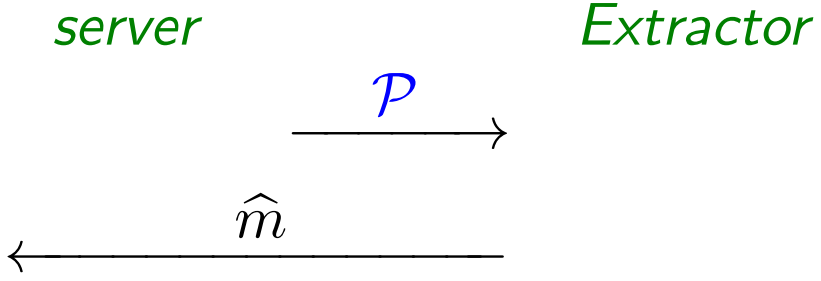
1. initialisation



2. audit



3. extraction



Our problem setting

- We study POR schemes in the setting of unconditional security, where the adversary is assumed to have unlimited computational capabilities.
- We only consider POR schemes where $\delta = 1$, that is, where extraction is guaranteed to be successful.
- The constructions that we utilise for extractors only require black-box access to the proving algorithm.
- In this setting, it turns out that extraction can be interpreted naturally as nearest-neighbour decoding in a certain code (which we term a response code).
- Error-correcting codes have been used in many constructions of POR schemes; we propose that error-correcting codes constitute the natural foundation to construct as well as analyse arbitrary POR schemes.

Why unconditional security?

- Simplicity and mathematical elegance: The schemes are mathematically elegant as well as easier to understand and analyse because we are not making use of any additional cryptographic primitives.
- Exact analyses: We can give very simple exact (i.e., non-asymptotic) analyses of various schemes.
- Links with error-correcting codes: The essential role of error-correcting codes in the design and analysis of POR schemes becomes clear: codes are not just a method of constructing POR schemes; rather, every POR scheme gives rise to a code in a natural way.
- Adversarial strength: It is interesting and informative to consider security against the strongest possible adversary and to prove security results that do not depend on unproven assumptions.

Basic Scheme

Initialisation

Given a message $m \in (\mathbb{F}_q)^k$, encode M as $e(m) = M \in (\mathbb{F}_q)^n$, where q is a prime power and $n \geq k$. The set of encoded messages is the encoded message space. We write $M = (m_1, \ldots, m_n)$.

Alice gives M to the *server*. *Alice* also generates a random challenge $c \in \{1, \ldots, n\}$ and she stores c and m_c.

Challenge-response

Alice gives the challenge c to the *server*. The *server* responds with $r = m_c$. *Alice* checks that $r = m_c$.

The extractor

1. Compute responses to all possible challenges: On input $\mathcal{P}$, compute the response vector $M' = (m'_1, \ldots, m'_n)$, where $m'_c = \mathcal{P}(c)$ for all $c \in \{1, \ldots, n\}$ (i.e., m'_c is the response from $\mathcal{P}$ when it is given the challenge c).
2. Nearest-neighbour decoding: Find an encoded message $\widehat{M}$ so that $\mathrm{dist}(M', \widehat{M})$ is minimised, where $\mathrm{dist}(\cdot, \cdot)$ denotes the hamming distance between two vectors.
3. Output $\widehat{m} = e^{-1}(\widehat{M})$.

Theorem

Suppose that $\mathcal{P}$ is a proving algorithm for the Basic Scheme for which

$$\mathrm{succ}(\mathcal{P}) > 1 - \frac{d}{2n},$$

where the hamming distance of the encoded message space is d. Then the Extractor will always output $\widehat{m} = m$.

Example

- Suppose that *Alice* wants to use the Basic Scheme with $q = 2^{10}$ and $n = 1000$ such that the minimum distance of the encoded message space is 400.
- This will guarantee that extraction will be possible whenever $\mathsf{succ}(\mathcal{P}) > 0.8$.
- If Alice uses a Reed-Solomon code to encode messages, then $d = n - k + 1$, where k is the dimension of the code.
- Therefore, $k = 601$, so the message expansion is

$$\frac{1000}{601} \approx 1.67.$$

- The size of a challenge is $\log_2 n = 10$ bits and the size of a response is $\log_2 q = 10$ bits.

Generalisation

We can consider arbitrary challenge-response protocols, where a challenge will be chosen from a specified challenge space Γ, and the response will be an element of a response space Δ. The response code consists of all $|\Gamma|$-tuples of elements from Δ that are obtained as correct responses for some encoded message M. We can prove a straightforward generalisation of the previous theorem.

Theorem

Suppose that $\mathcal{P}$ is a proving algorithm for a General POR Scheme for which

$$\mathsf{succ}(\mathcal{P}) > 1 - \frac{d^*}{2|\Gamma|},$$

where the hamming distance of the response code is d^*. Then the Extractor based on nearest neighbour decoding will always output $\widehat{m} = m$.

Multiblock Challenge Scheme

- Here, a challenge specifies ℓ indices "all at once", say $i_1 < \cdots < i_\ell$.
- $|\Gamma| = \binom{n}{\ell}$.
- The response is the ℓ-tuple $(m_{i_1}, \ldots, m_{i_\ell})$.
- If the hamming distance of the encoded message space is d, then the hamming distance of the response code is

$$d^* = \binom{n}{\ell} - \binom{n-d}{\ell}.$$

- Therefore, extraction succeeds if

$$\mathsf{succ}(\mathcal{P}) > \frac{1}{2} + \frac{\binom{n-d}{\ell}}{2\binom{n}{\ell}}.$$

Linear Combination Scheme

- A challenge V is a vector in $(\mathbb{F}_q)^n$ having hamming weight equal to ℓ.
- The response is

$$V \cdot M = \sum_{i=1}^{n} v_i m_i \bmod q.$$

- $|\Gamma| = \binom{n}{\ell}(q-1)^\ell$ and $|\Delta| = q$.
- If the hamming distance of the encoded message space is d, then a very accurate estimate for the hamming distance of the response code is

$$d^* \approx \frac{(q-1)^{\ell+1}}{q}\left(\binom{n}{\ell} - \binom{n-d}{\ell}\right).$$

- Therefore, extraction succeeds if

$$\mathrm{succ}(\mathcal{P}) > \frac{1}{2} + \frac{1}{2}\left(\frac{1}{q} + \frac{(q-1)\binom{n-d}{\ell}}{q\binom{n}{\ell}}\right).$$

Comparison

- The Linear Combination Scheme has much smaller responses than the Multiblock Challenge Scheme ($\mathbb{F}_q$ as opposed to $(\mathbb{F}_q)^\ell$).
- However, the Linear Combination Scheme has a larger challenge space than the Multiblock Challenge Scheme ($\binom{n}{\ell}(q-1)^\ell$ as opposed to $\binom{n}{\ell}$).
- The relative distance of the response codes of the two schemes are very similar, so the security guarantees of the two schemes are also very similar.

Example

- Suppose that *Alice* wants to use the Linear Combination Scheme with $q \geq 2^{10}$ and $n = 1000$.
- Her goal is that extraction will be possible whenever $\mathrm{succ}(\mathcal{P}) > 0.8$.
- Here, $d = 50$ and $\ell = 10$ will work.
- If Alice uses a Reed-Solomon code to encrypt messages, then $k = 951$, so the message expansion is

$$\frac{1000}{951} \approx 1.05.$$

- The size of a challenge is 178 bits and the size of a response is $\log_2 q = 10$ bits.

Estimating the success probability of a prover

- We have proven that extraction is possible provided that $\mathrm{succ}(\mathcal{P})$ is sufficiently close to 1.
- In general, the only way to determine the exact value of $\mathrm{succ}(\mathcal{P})$ is to query $\mathcal{P}$ with all the possible challenges (as is done during extraction).
- In practice, we would like to be able to estimate $\mathrm{succ}(\mathcal{P})$ based on a relatively small number of challenges.
- This can be done using classical statistical techniques such as hypothesis testing and confidence intervals.

Hypothesis testing for the Basic Scheme

- We know that extraction will be successful in the Basic Scheme if

$$\mathsf{succ}(\mathcal{P}) \geq \frac{n - \lfloor \frac{d}{2} \rfloor + 1}{n}.$$

- Denote $\omega = n - \lfloor \frac{d}{2} \rfloor + 1$.
- We wish to distinguish the null hypothesis

$$H_0 : \mathsf{succ}(\mathcal{P}) \leq \frac{\omega - 1}{n};$$

from the alternative hypothesis

$$H_1 : \mathsf{succ}(\mathcal{P}) \geq \frac{\omega}{n}.$$

- If we reject the null hypothesis H_0, then we believe that extraction is possible.

Hypothesis testing for the Basic Scheme (cont.)

- Suppose there are g correct responses in t trials.
- For simplicity, assume the challenges are chosen uniformly at random with replacement.
- The condition for rejecting the null hypothesis at a 5% significance level is

$$\sum_{i=g}^{t} \binom{t}{i} \left(\frac{\omega-1}{n}\right)^{i} \left(\frac{n-\omega+1}{n}\right)^{t-i} < 0.05.$$

- If this condition holds, then we are quite confident that successful extraction is possible.

Example

- Suppose that *Alice* using the Basic Scheme with $n = 1000$ and the minimum distance of the encoded message space is 400.
- Then extraction is possible whenever $\text{succ}(\mathcal{P}) > 0.8$.
- Suppose the *server* responds to 100 challenges that have been chosen uniformly with replacement, and that 87 of the responses were correct.
- We find that

$$\sum_{i=87}^{100} \binom{100}{i} 0.8^i \, 0.2^{100-i} \approx 0.047 < 0.05.$$

- There is sufficient evidence to reject the null hypothesis at the 5% significance level, and so we conclude that the file can be reconstructed by an extractor.

A new lower bound on storage and communication

- Suppose that $\mathbf{M}$ is a random variable corresponding to a randomly chosen unencoded message m.
- Let $\mathbf{V}$ be a random variable denoting any information stored by *Alice*
- Let $\mathbf{R}$ be a random variable corresponding to the information provided by a black-box *Extractor*.
- Suppose that the message can be reconstructed by the *Extractor* with probability 1
- Then

$$H(\mathbf{M}|\mathbf{V},\mathbf{R}) = 0,$$

from which it follows that

$$H(\mathbf{M}) \leq H(\mathbf{V}) + H(\mathbf{R}).$$

Lower bound (cont.)

- Naor and Rothblum proved a lower bound for a weaker form of POR-type protocol, termed an an authenticator.
- The Naor-Rothblum bound also applies to POR schemes.
- Phrased in terms of entropy, their bound states that

$$H(\mathbf{M}) \leq H(\mathbf{V}) \times H(\mathbf{R}),$$

  which is a weaker bound than the one we proved above.

Thank you for your attention!

Our results can be found in the preprint

A coding theory foundation for the analysis of general unconditionally secure proof-of-retrievability schemes for cloud storage

which will shortly appear on the IACR eprint archive. This preprint also contains a treatment of unconditionally secure keyed (i.e., unbounded-use) POR schemes.