Towards Efficient Proofs of Retrievability

Jia Xu
Institute for Infocomm Research
Singapore

Ee-Chien Chang
School of Computing
National University of Singapore
Problem: POR/PDP

Alice wants to periodically verify the integrity of a large file stored in a server Bob.

Performance
- Storage overhead
- Communication bits per verification
- \textit{Computation cost per verification}
- \textit{Setup Cost}

Oct 1, 2010
Our contributions

Shacham and Waters (Asiacrypt 08) gave a scheme with

$O(s\lambda)$ comm bits and overhead of $(1/s)$ file size, where $\lambda$ is a security parameter.

We improved the scheme to

$O(\lambda)$ comm bits and overhead of $(1/s)$ file size.
Background
Authenticator - Setup

Given the data, organized in blocks.

- Apply error erasure code on the data.
- For each block, compute an authentication tag (mac) using Alice’s secret key.
- Bob stores the encoded data and the tags.
- Alice only keeps her secret key.
Given the data, organized in blocks.

Apply error erasure code on the data.

For each block, compute an authentication tag (mac) using Alice’s secret key.

Bob stores the encoded data and the tags.

Alice only keeps her secret key.
Authenticator - Setup

- Given the data, organized in blocks.
- Apply error erasure code on the data.
- For each block, compute an authentication tag (mac) using Alice’s secret key.
- Bob stores the encoded data and the tags.
- Alice only keeps her secret key.
Authenticator - Setup

- Given the data, organized in blocks.
- Apply error erasure code on the data.
- For each block, compute an authentication tag (mac) using Alice’s secret key.
- Bob stores the encoded data and the tags.
- Alice only keeps her secret key.
Authenticator – Challenge/response

Challenge: Alice sends indices of blocks to be inspected

Response: Bob sends the blocks (messages & tags)
Verify: Alice verifies that the tags are valid

Security
If data stored in Bob’s storage is beyond reconstruction, the probability of a message randomly chosen block has invalid tag is more than 1/3 in the above example.

Oct 1, 2010
SW scheme-Setup

- Choose an appropriate field $\mathbb{Z}_p$
- The dataset is organized as vectors of $s$ elements ($s=3$ in the above e.g).
- For ease of comparison, let us ignore the step of using erasure code as in the Authenticator.

Oct 1, 2010
### SW scheme-Setup

<table>
<thead>
<tr>
<th>$s_1$</th>
<th>$s_2$</th>
<th>$s_3$</th>
<th>$s_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r^0 m_{1,1}$</td>
<td>$r^0 m_{1,2}$</td>
<td>$r^0 m_{1,3}$</td>
<td>$r^0 m_{1,4}$</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>$r^1 m_{2,1}$</td>
<td>$r^1 m_{2,2}$</td>
<td>$r^1 m_{2,3}$</td>
<td>$r^1 m_{2,4}$</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>$r^2 m_{3,1}$</td>
<td>$r^2 m_{3,2}$</td>
<td>$r^2 m_{3,3}$</td>
<td>$r^2 m_{3,4}$</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

- Alice computes a tag for each vector.
- Alice keeps the $r$ and $s_i$.
- Bob keeps tag $t_i$ and dataset $m_{i,j}$.

Oct 1, 2010
SW scheme-Setup

\begin{align*}
  s_1 & + \\
  s_2 & + \\
  s_3 & + \\
  s_4 & + \\
  f m_1(r) & \quad f m_2(r) & f m_3(r) & f m_4(r) \\
  t_1 & \quad t_2 & t_3 & t_4
\end{align*}
SW scheme-Challenge

• Alice sends $\nu_1, \nu_2, \nu_3, \nu_4$
Bob sends $\sigma$ and $\mu_1, \mu_2, \mu_3$
SW scheme-Verify

- Note that \( \sigma = f_{\mu}(r) + \langle \bar{s}, \bar{\nu} \rangle \) \hspace{1cm} (1)

- Verify whether the response is consistent with (1)
Remarks on Performance

- Reduction of communication bits required to indicate which vectors are to be used in the challenge
  - Random oracle
  - Dodis et al [13]

- The challenge can be replaced by $1, \nu, \nu^2, \nu^3$ and thus reduce the number of communication bits.

- The secret $s_i$ can be generated by a pseudorandom function, and thus reduce the size of keys.
Our Goal

Note that the response contains a vector

\[
\begin{bmatrix}
m_{1,1} \\
m_{2,1} \\
m_{3,1}
\end{bmatrix}
+ \begin{bmatrix}
m_{1,2} \\
m_{2,2} \\
m_{3,2}
\end{bmatrix}
+ \begin{bmatrix}
m_{1,3} \\
m_{2,3} \\
m_{3,3}
\end{bmatrix}
+ \begin{bmatrix}
m_{1,4} \\
m_{2,4} \\
m_{3,4}
\end{bmatrix}
= \begin{bmatrix}
\mu_1 \\
\mu_2 \\
\mu_3
\end{bmatrix}
\]

\[
\begin{bmatrix}
\nu_1 \\
\nu_2 \\
\nu_3 \\
\nu_4
\end{bmatrix}
\begin{bmatrix}
t_1 \\
t_2 \\
t_3 \\
t_4
\end{bmatrix}
= \sigma
\]

Our goal is to aggregate elements in the vector, and thus reduce the communication bits.
Proposed Scheme
Main idea

Using the fact:

For any polynomial \( f(x) \) and scalar \( r \),

\[(x-r) \text{ divides the polynomial } f(x) - f(r)\]

Kate et al [22] exploited the above to obtain constant size polynomial commitment scheme.
Setup

$s_1 + \tau \alpha^0 m_{1,1} + \tau \alpha^1 m_{2,1} + \tau \alpha^2 m_{3,1} \parallel t_1$

$s_2 + \tau \alpha^0 m_{1,2} + \tau \alpha^1 m_{2,2} + \tau \alpha^2 m_{3,2} \parallel t_2$

$s_3 + \tau \alpha^0 m_{1,3} + \tau \alpha^1 m_{2,3} + \tau \alpha^2 m_{3,3} \parallel t_3$

$s_4 + \tau \alpha^0 m_{1,4} + \tau \alpha^1 m_{2,4} + \tau \alpha^2 m_{3,4} \parallel t_4$

Oct 1, 2010
<table>
<thead>
<tr>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>( s_3 )</th>
<th>( s_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

\[
\tau f_{m_1}(\alpha) \quad \tau f_{m_2}(\alpha) \quad \tau f_{m_3}(\alpha) \quad \tau f_{m_4}(\alpha)
\]

<table>
<thead>
<tr>
<th>( t_1 )</th>
<th>( t_2 )</th>
<th>( t_3 )</th>
<th>( t_4 )</th>
</tr>
</thead>
</table>

Alice keeps \( \alpha \), \( \tau \), \( s_i \), \( g, g^\alpha, g^{\alpha^2} \)

Bob keeps \( t_i \), \( m_{i,j} \), \( g, g^\alpha, g^{\alpha^2} \)
Challenge

• Similarly, for the purpose of illustration, let’s ignore the selection of vectors and consider all vectors.

• Alice sends $\nu_1, \nu_2, \nu_3, \nu_4$ and a $r$
Bob divides \( f_{\bar{\mu}}(x) - f_{\bar{\mu}}(r) \) by \( (x - r) \) to obtain \( f_{\bar{w}}(x) \)

Together with \( g, g^\alpha, g^{\alpha^2} \), Bob can compute \( \psi = g^{f_{\bar{w}}(\alpha)} \)
The response consists of

\[ \psi = g^{f_{\bar{w}}(\alpha)} \]

\[ y = f_{\mu}(r) \]

\[ \sigma \]

Note that

\[ \sigma = \tau(f_{\mu}(\alpha) + \langle \bar{s}, \bar{\nu} \rangle) \]

\[ g^{f_{\mu}(\alpha)} - f_{\mu}(r) = g^{(\alpha - r)f_{\bar{w}}(\alpha)} \]  \[ \text{-------(2)} \]

Alice verifies the consistency of the responses with (1) & (2)
Remarks

• The $s_i$ can be generated using a PRF (pseudorandom function).

• Known methods on SW to reduce challenge size can be similarly applied.
Theorem

The proposed scheme is complete and sound POR scheme, assuming Strong Diffie Hellman (SDH) holds and the PRF is secure.
## Performance

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Group element size (bits)</th>
<th>Communication (bits)</th>
<th>Storage Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3][1]</td>
<td>$\lambda = 1024$</td>
<td>$2\lambda + 520$</td>
<td>$</td>
</tr>
<tr>
<td>[26]</td>
<td>$\lambda = 80$</td>
<td>$(s + 1)\lambda + 360$</td>
<td>$</td>
</tr>
<tr>
<td>EPOR (E.C.)</td>
<td>$\lambda = 160$</td>
<td>$3\lambda + 440$</td>
<td>$</td>
</tr>
<tr>
<td>EPOR ($\mathbb{Z}_q^*$)</td>
<td>$\lambda = 1024$</td>
<td>$3\lambda + 440$</td>
<td>$</td>
</tr>
</tbody>
</table>

Ateniese et al.
Shacham & Water
This paper (EC)
This paper
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computation (Prover)</th>
<th>Computation (Verifier)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3][1]</td>
<td>((\ell + s) \exp + 2\ell \text{ mult.} + \ell \text{ add} + 1 \text{ hash} + 1 \text{ samp})</td>
<td>(\ell \text{ (exp + mult.)} + 1 \text{ hash} + 1 \text{ samp})</td>
</tr>
<tr>
<td>[26]</td>
<td>(s\ell \text{ (add + mult)} + 1 \text{ samp})</td>
<td>((\ell + s) \text{ (add + mult)} + \ell \text{ PRF} + 1 \text{ samp})</td>
</tr>
<tr>
<td>EPOR (E.C.)</td>
<td>((s - 1) \exp + (s\ell + s + \ell) \text{ (add + mul)} + 1 \text{ samp})</td>
<td>(2 \exp + \ell \text{ (add + mult)} + \ell \text{ PRF} + 1 \text{ samp})</td>
</tr>
<tr>
<td>EPOR ((\mathbb{Z}_q^*))</td>
<td>((s - 1) \exp + (s\ell + s + \ell) \text{ (add + mul)} + 1 \text{ samp})</td>
<td>(2 \exp + \ell \text{ (add + mult)} + \ell \text{ PRF} + 1 \text{ samp})</td>
</tr>
</tbody>
</table>

Ateniese et al.  
Shacham & Water  
This paper (EC)  
This paper  

Oct 1, 2010
<table>
<thead>
<tr>
<th>Scheme</th>
<th>Computation (Data Preprocess)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3][1]</td>
<td>$</td>
</tr>
</tbody>
</table>
| [26]         | $|F|/\lambda (\text{mul } + \text{ add}) + $  
|              | $|F|/((\lambda s) \text{ PRF})$ |
| EPOR (E.C.)  | $|F|/\lambda (\text{mul } + \text{ add}) + $  
|              | $|F|/((\lambda s) \text{ PRF})$ |
| EPOR ($\mathbb{Z}_q^*$) | $|F|/\lambda (\text{mul } + \text{ add}) + $  
|              | $|F|/((\lambda s) \text{ PRF})$ |

Ateniese et al.
Shacham & Water
This paper (EC)
This paper
Conclusion

• We incorporate idea of Kate et al. to reduce the communication cost of SW scheme.

• Compare to the scheme by Ateniese et al, the proposed scheme has similar communication cost and storage overhead, but require less computation cost during verification and setup.