The Merkle-tree Lemma

Christopher Smith

Last Updated: January 8, 2024

A self-contained document on the statement and proof of the Merkle-tree lemma, as found in [1].

1 Definitions

Definition 1.1 (Merkle Tree)**.** Denote by $MT_{h,b}(X)$ the *Merkle tree* over string $X \in \{0,1\}^*$ with hash function h and b-bit leaf values. For each node $n \in MT_{h,b}(X)$, denote by v_n the value associated with node n. The value of a leaf is the corresponding block of X, and the value of an intermediate node n is the hash $v_n = h(v_l, v_r)$, where v_l and v_r are the values of the left and right children of n. $MT_{h,b}(X)$ is a completely balanced binary tree, as we can fill in missing nodes with empty string valued nodes.

Definition 1.2 (Sibling Path)**.** For a leaf node $l \in MT_{h,b}(X)$, the *sibling path of* l consists of the value v_l, along with all the values of all the siblings of nodes on the path from l to the root.

Definition 1.3 (Valid Path)**.** An alleged sibling path $(v_l, v_{n_0}, ..., v_{n_i})$ is *valid with respect to* $MT_{h,b}(X)$ if i is the height of the tree, and the root value as computed on the sibling path agrees with the root value of $MT_{h,b}(X)$.

Note: In order to verify a given alleged sibling path, it suffices to know the hash h, the number of leaves, and the root value of $MT_{h,b}(X)$.

Definition 1.4 (Merkle-tree Protocol)**.** Denote by $MTP_h(v, s, u)$ the *Merkle-tree Protocol* with respect to hash function h where the verifier knows the root value v and number of leaves s, and asks the prover to see q leaves of the tree along with sibling paths. The verifier accepts if all the sibling paths are valid.

2 Lemma and Proof

Lemma 2.1 (Merkle-tree Lemma)**.** *There exists a black-box extractor K with oracle access to a Merkle-tree prover, that has the following properties:*

1. *For every prover P and $v \in \{0,1\}^*$, $s, u \in \mathbb{N}$, and $\delta \in [0,1]$, $K^P(v, s, u, \delta)$ makes at most $u^2 s(\log(s)+1)/\delta$ calls to its prover oracle P.*
2. *Fix any hash function h and string $X \in \{0,1\}^{sb}$, and let v be the root value of $MT_{h,b}(X)$. Also fix some $u \in \mathbb{N}$, and a prover P^* that may depend on h, X, u.*
   Then if P^* has probability at least $(1-\alpha)^u + \delta$ of convincing the verifier in the Merkle-tree protocol $MTP_h(v, s, u)$ (for some $\alpha, \delta \in (0,1]$), then with probability at least 1/4 (over its internal randomness) the extractor $K^{P^*}(v, s, u, \delta)$ outputs values for at least $(1-\alpha)s$ of the leaves, together with valid sibling paths for all these leaves.

Proof. Let $\alpha, \delta \in (0,1]$, $u \in \mathbb{N}$. Fix a hash function h, and a string $X \in \{0,1\}^{sb}$. Fix a prover P^* that possibly depends on h, X, u, and suppose P^* convinces the verifier in $MTP_h(v, s, u)$ with probability at least $(1-\alpha)^u + \delta$.

Consider the following extractor K:

procedure $K^{P*}(v, s, u, \delta)$
    for $i = 1$ to u **do**
        for $l = 1$ to s **do**
            for $u(\log(s)+1)/\delta$ times **do**
                Choose at random $l_1, ..., l_u \in [s]$
                Query $P^*(l_1, ..., l_{i-1}, l, l_{i+1}, ..., l_u)$
            end for
        end for

end for
Output sibling paths for all the leaves for which P^* ever gave a valid sibling path.
end procedure

It is obvious that K makes at most $u^2 s(\log(s)+1)/\delta$ calls to its oracle, so property 1 of the lemma is satisfied. We must now show that K outputs values for at least $(1-\alpha)s$ of the leaves with probability at least $1/4$.

We say "$P^*(l_1,...,l_u)$ is valid" if P^* responds with valid sibling paths for every leaf when queried on leaves $l_1,...,l_u$.

For leaf index $l \in [s]$ and query index $i \in [u]$, we say "*l is i-good*" if $\Pr_{l_1,...,l_u}[P^*(l_1,...,l_{i-1},l,l_{i+1},...,l_u) \text{ is valid}] \geq \delta/u$.

For $i \in [u]$, let $\mathrm{Good}_i := \{l \in [s] : l \textit{ is } i\textit{-good}\}$.

A key claim is that there exists at least one query index $\hat{i} \in [u]$ such that $|\mathrm{Good}_{\hat{i}}| \geq (1-\alpha)s$. To prove the claim, assume for contradiction that $\forall i \in [u]$, $|\mathrm{Good}_i| < (1-\alpha)s$. Then we have:

$$\begin{aligned}
&\Pr_{l_1,\ldots,l_u}[P^*(l_1,...,l_u) \text{ is valid}] \\
&\quad= \Pr[P^*(l_1,...,l_u) \text{ is valid AND } l_i \text{ is } i\text{-good } \forall i \in [u]] \\
&\quad+ \Pr[P^*(l_1,...,l_u) \text{ is valid AND } \exists i \in [u] : l_i \text{ is not } i\text{-good}] \\
&\quad\leq \Pr\left[\bigcap_{i\in[u]} l_i \text{is } i\text{-good}\right] + \Pr\left[P^* \text{is valid AND } \bigcup_{i\in[u]} l_i \text{is not } i\text{-good}\right] \\
&\quad\leq \prod_{i=1}^{u} \frac{|\mathrm{Good}_i|}{s} + \sum_{i=1}^{u} \Pr[P^* \text{ is valid} \mid l_i \text{is not } i\text{-good}] \\
&\quad< (1-\alpha)^u + u(\delta/u) = (1-\alpha)^u + \delta
\end{aligned}$$

But this is a contradiction since $\Pr_{l_1,\ldots,l_u}[P^*(l_1,...,l_u)\text{is valid}] \geq (1-\alpha)^u + \delta$ by assumption. Thus the claim holds.

Now consider the inner loop of the extractor code with some i, l, where l is i-good. Let $X_{i,l}$ be the binomial random variable for the number of times $P^*(l_1,...,l_{i-1},l,l_{i+1},...,l_u)$ is valid in the inner loop. We have:

$$\begin{aligned}
&\Pr[X_{i,l} = 0] \\
&\quad< \left(1 - \frac{\delta}{u}\right)^{u(\log(s)+1)/\delta} \\
&\quad< (e^{-\delta/u})^{u(\log(s)+1)/\delta} && ((1+x) \leq e^x) \\
&\quad< e^{-\log(s)-1} \\
&\quad\implies \Pr[X_{i,l} \geq 1] \geq 1 - e^{-\log(s)-1} \geq 1 - \frac{1}{es}
\end{aligned}$$

Let X_i be the binomial random variable for the number of times that P^* is valid at least once in the innermost loop when l is i-good. When K reaches index $\hat{i}$ in the outer loop we get that:

$$\begin{aligned}
&\Pr[X_{\hat{i}} \geq (1-\alpha)s] \\
&\quad\geq \left(1 - \frac{1}{es}\right)^{(1-\alpha)s} \\
&\quad\geq \left(1 - \frac{1/e}{s}\right)^{s} \\
&\quad\geq \left(1 - \frac{1}{e}\right) > 1/4 && ((1+x/n)^n \geq 1+x)
\end{aligned}$$

□

Note: If h is collision-resistant, then valid query responses are consistent with the original input string X.

References

[1] Shai Halevi, Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg. Proofs of ownership in remote storage systems. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, page 491–500, New York, NY, USA, 2011. Association for Computing Machinery. `https://eprint.iacr.org/2011/207`.