Leaf-Binding in a 2-Leaf Merkle Tree: A Lean 4 Proof

Risiraj Dey

04-04-2026

Overview

This document proposes a way to formally verify the leaf-binding property of a single-level, 2-leaf Merkle tree in Lean 4. Specifically, it is proved that any signature which passes tree verification commits to exactly one of the two leaf public keys, determined by the index bit. Hash values are typed as `BitVec 256`, matching the output width of SHA-256, with the hash function H and its injectivity assumption bundled in a `HashModel` typeclass.

Core Property

The main theorem proved is `leaf_binding`: given a valid signature σ under root $H(pk_0, pk_1)$, the index bit determines the leaf uniquely:

$$\mathit{verify}\ lv\ (H\, pk_0\, pk_1)\ m\ \sigma \implies (\sigma.\mathit{idx} = \texttt{false} \wedge \sigma.\mathit{leaf\_pk} = pk_0) \vee (\sigma.\mathit{idx} = \texttt{true} \wedge \sigma.\mathit{leaf\_pk} = pk_1)$$

where *verify* checks an abstract leaf predicate lv and the tree equation:

$$\big(\text{if } \sigma.\mathit{idx} \text{ then } H(\sigma.\mathit{sibling},\, \sigma.\mathit{leaf\_pk}) \text{ else } H(\sigma.\mathit{leaf\_pk},\, \sigma.\mathit{sibling})\big) = r$$

A compositional theorem `compositional` additionally combines tree binding with the leaf predicate, yielding both the index commitment and $m = \sigma.\mathit{leaf\_pk}$ as a single result. Forgery rejection is proved: a signature with an incorrect *leaf_pk* cannot satisfy the tree equation, closed by `H_inj` and `simp` on the concrete `BitVec 256` keys.

Challenges

1. **`H_inj` is an unproved axiom; tree equations close definitionally.** H and its injectivity assumption $H(a, b) = H(c, d) \Rightarrow a = c \wedge b = d$ are introduced via the `HashModel` typeclass with no concrete instance. Unlike the prior version, the concrete tree equations $H(pk_0, pk_1) = \mathit{root}$ are *not* separately axiomatized — they close by `simp` unfolding the signature struct fields definitionally, since $\sigma.\mathit{leaf\_pk} = pk_0$ and $\sigma.\mathit{sibling} = pk_1$ hold by construction. The only remaining axiom is `H_inj` itself, which is the standard assumption underlying all Merkle tree binding proofs.

2. **Scope is strictly structural.** The `leaf_ver` predicate is defined as $m = \mathit{leaf\_pk}$, a black-box equality with no cryptographic content. The theorem says nothing about what the leaf contains or how it was produced — only that the tree equation pins the index to one leaf.

3. **Tree height is fixed at 1.** The formalization covers exactly $\mathit{root} = H(pk_0, pk_1)$. Generalizing to height h requires `Vector HashVal h` for the authentication path, `Fin`(2^h) for the leaf index, a recursive root recomputation via `foldl`, and an inductive proof applying `H_inj` at each level — all outside the scope of this work.