Scroll-revm & zkVM Guest Program Changes — Security Assessment

Authors
- Rohit Narurkar
- Huaihuaqing Zhang
- Zhuo Zhang
- Souhail Mssassi

Date: 21 July 2025

Executive Summary

Severity	Count	Identifiers	Potential Impact
Informational	9	I-1 → I-9	Code-hygiene and refactor items; addressing them improves robustness and future maintainability.

Scope
- **Scroll REVM** — https://github.com/scroll-tech/scroll-revm/tree/6c1942f9a8eaf7aae1807654c4ee99d771150fbd
- **zkVM Guest Program Changes (precompiles)** — https://github.com/scroll-tech/stateless-block-verifier/commit/daeeb9e193bbb7e3a0438dd823b3b6c3310775ea

---

1 · Introduction

This audit covers two tightly–coupled codebases that implement the **Feynman upgrade** for Scroll L2.
Supporting documents are available here:
https://www.notion.so/Feynman-Upgrade-Documents-Shared-with-Auditors-2077792d22af804bae69ce529aa770f3
All findings were identified against these frozen commits; line numbers map exactly.

---

2 · Findings

2.1 Panics from unchecked L1 fee fields (I-1)

Severity: Informational
File Impacted: `src/l1block.rs` (L150-171 · 212-230 · 263-265) — *scroll-revm*

Description `data_gas`, `calculate_tx_l1_cost_curie`, and `calculate_tx_l1_cost_feynman` dereference optional fee-related fields via `unwrap` / `expect`.
Trigger vectors

1. **Cold start**: node boots with empty DB → `L1BlockInfo::default()` has `None` fields.

2. **Corrupted storage**: partial writes or replay gaps leave fields unset.

3. **Mixed hard-fork setup**: Curie/Feynman fee fields absent on older replica.

Any transaction using such a node **panics** the runtime → validator crash-loop or stalled RPC service.

Recommendation / Fix

```
#[derive(Debug, thiserror::Error)]
pub enum FeeError {
    #[error("uninitialised L1 fee field: {0}")]
    Uninitialised(&'static str),
    #[error("compression ratio below precision")]
    InvalidCompressionRatio,
}

fn data_gas(&self, input: &[u8]) -> Result<U256, FeeError> {
    let base   = self.l1_blob_base_fee.ok_or(FeeError::Uninitialised("l1_blob_base_fee"))?;
    let scalar = self.l1_blob_scalar   .ok_or(FeeError::Uninitialised("l1_blob_scalar"))?;
    Ok(U256::from(input.len()).saturating_mul(base).saturating_mul(scalar))
}
```

2.2 Assertion on compression ratio causes DoS (I-2)

Severity: Informational
File Impacted: `src/l1block.rs` (L206-210) — *scroll-revm*

Description `calculate_tx_l1_cost_feynman` asserts `compression_ratio` `1 000 000 000`. A crafted transaction with `compression_ratio = 0` trips the assertion → process abort.

Recommendation / Fix

```
if compression_ratio < TX_L1_FEE_PRECISION_U256 {
    return Err(FeeError::InvalidCompressionRatio);
}
```

- Update callers to handle `Result`.

2.3 Non-thread-safe precompile cache (I-3)

Severity: Informational
File Impacted: `src/precompile/mod.rs` (L54-102) — *scroll-revm*

Description The cache uses `once_cell::race::OnceBox` (unsynchronised). Concurrent EVM instantiation can double-allocate or expose partially initialised data.

Recommendation / Fix

```
use once_cell::sync::OnceBox;           // thread-safe
static INSTANCE: OnceBox<Precompiles> = OnceBox::new();

pub fn precompiles() -> &'static Precompiles {
    INSTANCE.get_or_init(|| Precompiles::for_spec(ScrollSpecId::Feynman))
}
```

2.4 Missing `#![forbid(unsafe_code)]` (I-4)

Severity: Informational
File Impacted: `src/lib.rs` — *scroll-revm*

Description The project currently contains no `unsafe` blocks, but does not forbid them; future contributors might add unsound code unnoticed.

Recommendation / Fix

```
// src/lib.rs
#![forbid(unsafe_code)]
```

Enforce via CI (`cargo deny`) to reject any new `unsafe` usage.

2.5 System-TX validation and accounting foot-guns (I-5 → I-8)

Severity: Informational
Files Impacted: `src/handler.rs` — *scroll-revm*

ID	Risk	Root Cause	Fix
I-5.1	Misconfigured system TX fails in `validate_against_state_and_deduct_caller`	gas_price/basefee 0	skip call when `is_system_tx`
I-5.2	Balance divergence in `reward_beneficiary`	same field misuse	bypass reward for system TX
I-5.3	Redundant refund logic	post-exec refund unnecessary	add `is_system_tx` guard

All three are *operational* hazards rather than exploitable bugs.

2.6 Manual hard-fork checks clutter instruction handlers (I-5.4)

Severity: Informational
File Impacted: `src/instructions.rs` — *scroll-revm*

Create macro `ensure_hf!` to match upstream `revm` style; reduces merge conflicts.

2.7 `L1BlockInfo` refactor for panic-free ergonomics (I-5.5)

Severity: Informational
File Impacted: `src/l1block.rs` — *scroll-revm*

Split era-specific fields into nested structs; remove 35 `unwrap!` calls;

2.8 `ScrollSpecId` default variant should be `Feynman` (I-5.6)

Severity: Informational
File Impacted: `src/spec.rs` — *scroll-revm*

Change `Default` impl; add compile-time lints for inadvertent default construction.

2.9 Handle trivial case in `encode_g1_point` (I-9)

Severity: Informational
File Impacted: `crates/precompiles/src/imps/bn128/openvm.rs` — *stateless-block-verifier*

Description Point-at-infinity encodes by zero-copy; current impl needlessly reverses 64 zero-bytes.

Recommendation / Fix

```
#[inline]
pub(super) fn encode_g1_point(p: G1Affine) -> [u8; G1_LEN] {
    let mut out = [0u8; G1_LEN];
    if !p.is_identity() {
        let (x, y) = (p.x().as_le_bytes(), p.y().as_le_bytes());
        for i in 0..FQ_LEN {
            out[i]          = x[FQ_LEN - 1 - i];
            out[i + FQ_LEN] = y[FQ_LEN - 1 - i];
        }
    }
    out
}
```