Wrong calculation of supply debt balance of a position in Lending protocol

---

What is the Supply/Debt Balance in Lending Protocols?

In DeFi lending protocols like Aave, Compound, or Venus:

- **Supply Balance**: Tracks how much a user has deposited (including accrued interest).
- **Debt Balance**: Tracks how much a user owes (also including accrued interest).

These balances **aren’t stored as raw values**. Instead, they’re often **stored as "scaled" balances**, and updated **lazily**, using **indexes** or **interest rate models**.

What Happens with Wrong Balance Calculations?

If Supply Balance is Miscalculated:

- Users might **receive more interest than they should**.
- The protocol might think it's more solvent than it is.
- It could lead to **over-distribution of interest** or **liquidity mismanagement**.

If Debt Balance is Miscalculated:

- Users may **not be liquidated on time → bad debt builds up**.
- Or they might be **liquidated too early**, causing unfair losses.
- Flash loan and recursive borrow exploits might become possible.

Common Causes of Wrong Balance Calculations

Root Cause	Description
Incorrect Index Math	Errors in applying the interest index when updating balances.
Improper Timestamp Handling	Skipping or miscalculating time deltas in interest accrual.
Precision Errors	Integer division, rounding, or overflows in token math.
Double Counting Accruals	Applying interest multiple times or not resetting indexes correctly.
Debt Repayment Bugs	Failing to update debt indexes on repayment → user stays in debt falsely.

Root Cause	Description
Wrong Scaling Factors	Confusing WAD (1e18) with RAY (1e27) precision in protocols like Maker/Aave.
Stale Oracle Prices	Debt vs collateral is calculated using old prices, leading to liquidation issues.

Real Examples

1. **Compound Finance (2021)**
   A bug in the `Drip()` function of their `Comptroller` led to overdistribution of COMP rewards. While not exactly a debt balance issue, it stemmed from miscalculated indexes.
2. **Iron Bank x Alpha Homora Exploit (2021)**
   Recursive borrow/lend between protocols led to huge amounts of uncollateralized borrowing because balances weren't correctly tracked across systems.
3. **C.R.E.A.M. Finance Hacks**
   Improper index updating and oracle mismanagement led to massive under-collateralized loans.

Where to Look in Code

In protocols like Aave/Compound, inspect:

- `accrueInterest()` — core to debt calculation.
- `getUserAccountData()` — reports collateral, debt, and health factor.
- `updateIndexes()` or `calculateBalance()` — functions updating scaled balances.
- `borrow()`, `repay()`, `liquidate()`, `withdraw()` — ensure index application is consistent.

Look at how:

```
debt = principal * borrowIndex / lastBorrowIndex
```

or:

```
supply = principal * supplyIndex / lastSupplyIndex
```

is handled — these calculations are easy to mess up.

How to Prevent/Mitigate This

Accurate Index Management :

- **Use separate indexes** for supply and borrow (e.g., `supplyIndex`, `borrowIndex`) and **update them correctly** whenever interest accrues.
- **Use fixed-point math libraries** (e.g. OpenZeppelin's `SafeMath`, `ABDKMathQuad`).
- **Formal verification** for index update logic.
- **Unit tests with time deltas**, interest accrual simulations.
- Use **fuzz testing tools** like Echidna to test balance edge cases.