#11 Lending

Lecture Notes for CS190N: Blockchain Technologies and Security November 5, 2025

This lecture explores the architecture of decentralized lending protocols, examining how DeFi replaces traditional banking intermediaries with smart contracts, the role of over-collateralization, algorithmic interest rates, automated liquidation mechanisms, and the critical oracle problem. A case study of Aave illustrates these concepts in practice.

1 THE TRUSTLESS BANKING PROBLEM: LENDING TO STRANGERS

Traditional finance (TradFi) relies on centralized intermediaries like banks to arbitrate trust. Banks assess risk through credit history and identity verification (KYC) and use legal contracts for recourse in case of default. This system is inherently permissioned.

Decentralized Finance (DeFi) replaces these intermediaries with self-executing smart contracts on a public blockchain. This creates an open, permissionless system where users interact directly with code, substituting institutional trust with cryptographic certainty. This model eliminates the need for credit checks or a central authority, requiring only a crypto wallet and an internet connection.

However, smart contracts are rigid and cannot access off-chain data or initiate legal action. This lack of traditional recourse is the central design challenge for DeFi lending. The system must safely lend to anonymous users, which necessitates a risk management architecture based entirely on on-chain, verifiable assets. This foundational constraint leads directly to the principle of over-collateralization.

2 THE DIGITAL PAWN SHOP: OVER-COLLATERALIZATION

DeFi lending operates like a global, automated pawn shop. A borrower receives a loan by pledging a valuable digital asset as collateral, and the protocol does not need to know the borrower's identity because the loan is secured by the collateral itself.

2.1 The Mechanics of Over-collateralization

Over-collateralization is the practice of requiring a borrower to pledge collateral worth more than the loan they receive. This creates a safety buffer against the price volatility of crypto assets, protecting the protocol from "bad debt" [1, 2].

The key metric is the **Loan-to-Value (LTV) ratio**, which defines the maximum amount a user can borrow against a specific collateral asset. For example, with an 80% LTV for Ethereum (ETH), a user depositing $1,000 of ETH can borrow up to $800 of another asset. LTV ratios are set by protocol governance and are lower for more volatile assets [1].

While effective, this model is impractical for those without significant capital to use as collateral. Its primary users are crypto-native participants who use it for on-chain strategies like:

- **Leveraged Trading:** Borrowing stablecoins against ETH to buy more ETH.
- **Accessing Liquidity:** Borrowing against assets to avoid selling them and triggering a taxable event.

3 THE ON-CHAIN THERMOSTAT: ALGORITHMIC INTEREST RATES

In DeFi, interest rates are not set by a central authority but are determined algorithmically based on real-time market conditions. The core driver is the **Utilization Rate (U)**.

The Utilization Rate is the percentage of a liquidity pool's assets that is currently being borrowed [2]:

$$U = \frac{\text{Total Borrowed}}{\text{Total Available Liquidity}}$$

It acts as a real-time supply and demand signal. When U is low, interest rates fall to encourage borrowing. When U is high, interest rates rise to incentivize repayment and attract new liquidity.

3.1 The Kinked Interest Rate Model

To balance capital efficiency with liquidity risk, protocols like Aave use a **"kinked" interest rate model**. The interest rate curve has two slopes, with a sharp "kink" at an **Optimal Utilization Rate ($U_{optimal}$)**, typically around 80-90% [1, 2].

- **Below U_{optimal} (Gentle Slope):** Interest rates rise slowly to encourage borrowing and maximize capital efficiency.
- **Above U_{optimal} (Steep Slope):** Interest rates rise sharply. This acts as a circuit breaker to discourage further borrowing, preventing the pool from being drained, and to attract new liquidity with high returns.

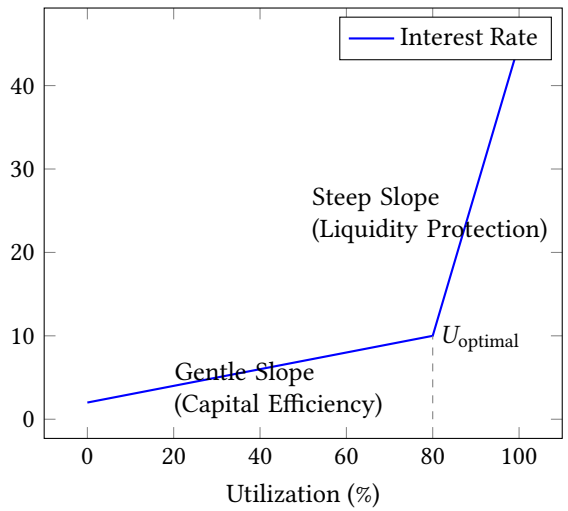

Fig. 1. The Kinked Interest Rate Curve. The interest rate rises slowly until the optimal utilization point is reached, after which it increases sharply to discourage further borrowing and protect the pool's liquidity.

4 THE IMMUNE SYSTEM: AUTOMATED LIQUIDATION

4.1 The Health Factor: A Position's Safety Metric

The **Health Factor (HF)** is the primary metric used by protocols like Aave to quantify the safety of a borrower's position and determine its proximity to a liquidation event. It is a real-time, on-chain value that measures the ratio of the total value of a borrower's collateral to their total outstanding debt, adjusted for the collateral's specific risk parameters.

The formula for the Health Factor is:

$$\text{HF} = \frac{\text{Total Collateral Value} \times \text{Weighted Average Liquidation Threshold}}{\text{Total Borrow Value}} \tag{1}$$

A health factor above 1 indicates a safe position that is well above the liquidation threshold. Conversely, when the health factor falls to 1 or below, the position is deemed under-collateralized and becomes eligible for liquidation. This can occur when the value of the collateral asset decreases, the value of the borrowed asset increases, or the borrower takes on more debt. To manage their risk, borrowers must monitor their health factor and can improve it by either supplying more collateral or repaying part of their loan.

Let's consider a concrete example using the Aave protocol's model. Suppose a user supplies $10,000 in ETH as collateral with a liquidation threshold of 80%. They then borrow $6,000 in GHO (a stablecoin). Their Health Factor is calculated as:

$$\text{HF} = \frac{\$10,000 \times 0.80}{\$6,000} \approx 1.333 \tag{2}$$

A health factor of 1.333 is above the critical threshold of 1, indicating a safe position. However, if the price of ETH drops and the collateral value falls to $7,000, the health factor would become:

$$\text{HF} = \frac{\$7,000 \times 0.80}{\$6,000} \approx 0.933 \tag{3}$$

Since the health factor is now below 1, the position is under-collateralized and is at risk of being liquidated.

4.2 The Liquidation Process

Liquidation is an automated process executed by external participants called **liquidators**, who are incentivized by profit.

(1) **Trigger:** A borrower's Health Factor falls below 1.
(2) **Detection:** Liquidators running automated bots detect the vulnerable position.
(3) **Execution:** A liquidator calls the protocol's `liquidate()` function, repaying the borrower's debt.
(4) **Incentive:** In return, the liquidator claims a portion of the borrower's collateral at a discount, known as the **liquidation bonus**.

Let's apply this to our example from Section 4.1, where the HF dropped to 0.933 ($7,000 ETH collateral, $6,000 GHO debt). Assume the protocol offers a **5% liquidation bonus**.

- **Action:** A liquidator steps in and repays the borrower's $6,000 GHO debt to the protocol.
- **Incentive:** In return, the liquidator is allowed to seize $\$6,000 \times 1.05 = \$6,300$ worth of the borrower's ETH collateral.
- **Who gets what?:**
  - **The Liquidator:** Spends $6,000 GHO and receives $6,300 in ETH, making a **$300 profit**.
  - **The LPs (Protocol):** Get their $6,000 GHO back, and the protocol remains solvent.
  - **The Borrower:** Their debt is cleared, and they are left with the remaining collateral: $\$7,000 - \$6,300 =$ **$700** in ETH.

This open market for risk mitigation ensures the protocol remains solvent. However, during a market crash, a wave of simultaneous liquidations can create a "death spiral," where forced selling of collateral further depresses asset prices, triggering more liquidations [1].

5 THE ORACLE DILEMMA: BRIDGING TO THE REAL WORLD

Blockchains are isolated systems and cannot access external, real-world data like asset prices. This is the **"Oracle Problem"**. A **blockchain oracle** is a service that feeds external data to

smart contracts [3]. For lending protocols, oracles are critical for valuing collateral and triggering liquidations; robust designs often aggregate multiple sources and/or use on-chain time-weighted pricing [4].

5.1 The Flash Loan Oracle Manipulation Attack

A **flash loan** is a unique DeFi mechanism that allows a user to borrow massive amounts of assets with zero collateral, under one critical condition: the loan must be borrowed and repaid within the **same transaction**.

This is possible because blockchain transactions are **atomic**, they either complete entirely (all steps succeed) or fail entirely (all steps are reverted), as if the transaction never happened. A flash loan is just one step in a larger, complex transaction. The lending smart contract checks at the very end of the transaction: ”Did I get my money back?” If the answer is no, the contract reverts the entire transaction, including the initial loan, making the loan risk-free for the lender.

This powerful ”money lego” is a tool for arbitrage and liquidations, but it also creates a new attack surface. A protocol’s reliance on oracles becomes a prime vulnerability when combined with the massive, temporary capital from a flash loan. Figure 2 illustrates the flow of the attack.

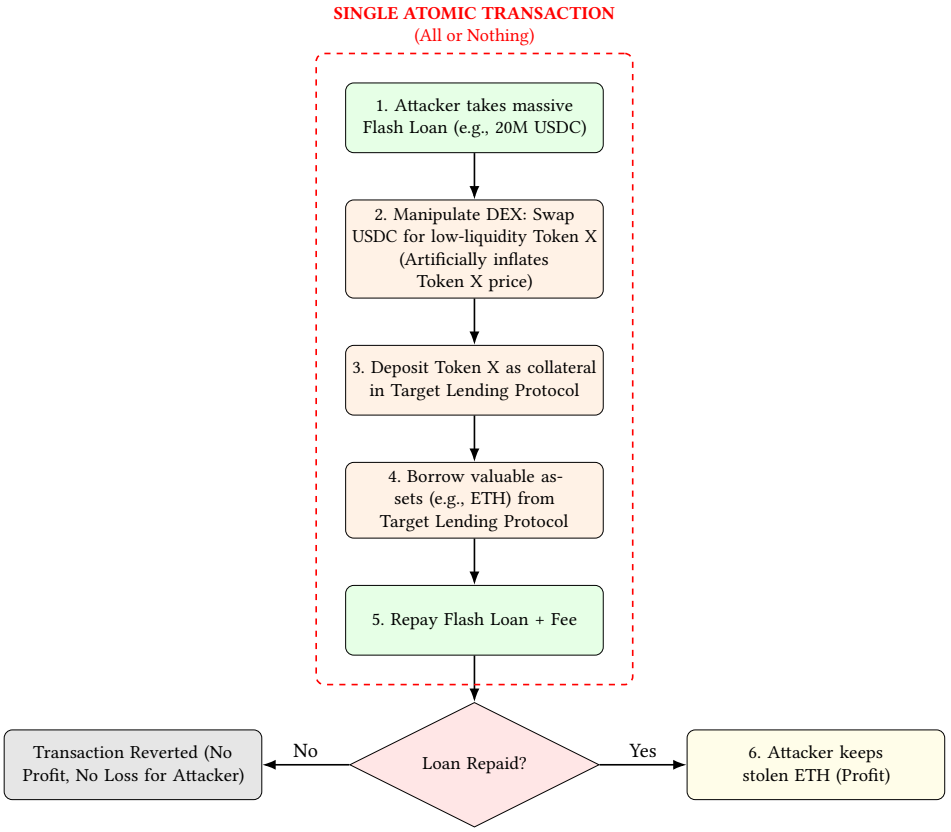

Fig. 2. Flow of a flash loan oracle attack. Steps 1-5 must all execute successfully within a single, atomic blockchain transaction. If the flash loan repayment (Step 5) fails, the entire transaction is reverted, protecting the flash loan lender.

An attack typically unfolds as follows:

(1) **Borrow:** An attacker takes a massive flash loan of a stablecoin (e.g., 20M USDC).
(2) **Manipulate:** They use the funds to execute a huge swap on a decentralized exchange (DEX), artificially inflating the price of a low-liquidity token (Token X) on that DEX.
(3) **Exploit:** The attacker deposits the artificially inflated Token X as collateral into a lending protocol that uses the manipulated DEX price as its oracle. They then borrow a large amount of a valuable asset (e.g., ETH).
(4) **Repay & Profit:** The attacker repays the flash loan and keeps the stolen ETH as profit. The lending protocol is left with worthless collateral and a large bad debt.

Here is a more concrete example. The attacker executes all of the following steps within a single, atomic transaction:

- **Initial State:** A "Target Lending Protocol" has $5M of ETH. It uses a small, low-liquidity DEX as its price oracle for "TokenX". On this DEX, 1 TokenX = $1. The protocol's LTV for TokenX is 75%.
- **Step 1 (Flash Loan):** The attacker **takes a 2M USDC flash loan** from Aave. (This is the massive, uncollateralized loan that must be repaid by the end of this transaction).
- **Step 2 (Manipulate):** The attacker uses the 2M USDC to buy TokenX on the small DEX. This massive buy order overwhelms the liquidity, and the oracle's spot price for TokenX skyrockets from $1 to $50.
- **Step 3 (Exploit - Deposit):** The attacker deposits 100,000 TokenX (which they held previously) into the Target Lending Protocol. The protocol's oracle, reading the manipulated DEX price, values this collateral at: $100,000 \text{ TokenX} \times \$50/\text{TokenX} = \mathbf{\$5{,}000{,}000}$.
- **Step 4 (Exploit - Borrow):** Using this inflated collateral, the attacker **borrows** $3.75M worth of ETH from the Target Protocol (this is the *second*, collateralized loan, which is the actual theft).
- **Step 5 (Repay & Profit):** The attacker repays their initial 2M USDC flash loan (plus a small fee) to Aave.

The Aftermath: The single transaction completes successfully because the flash loan (Step 1) was repaid (Step 5). The attacker has successfully stolen $3.75M in ETH. The price of TokenX on the DEX immediately crashes back to $1. The Target Protocol is now insolvent: it is missing $3.75M of its ETH, and in its place, it holds collateral (100,000 TokenX) worth only $100,000, leaving a bad debt of $3.65M.

To mitigate this, secure protocols use **Decentralized Oracle Networks (DONs)** like Chainlink, which aggregate data from many sources, or **Time-Weighted Average Price (TWAP)** oracles, which are resilient to short-term price spikes.

6 CASE STUDY: AAVE

Aave is a leading decentralized, non-custodial liquidity protocol that has become a cornerstone of the DeFi ecosystem. It is known for its feature-rich and flexible design, which caters primarily to advanced users and developers seeking maximum capital efficiency.

Key innovations and features of Aave include:

- **Flexible Interest Rates:** Unlike protocols that offer a single rate model, Aave provides users with a choice between variable and stable interest rates, allowing for more sophisticated risk management strategies.
- **Flash Loans:** Aave pioneered the concept of flash loans, which are uncollateralized loans that must be borrowed and repaid within the same blockchain transaction [1]. This powerful

tool has become a fundamental “money lego” for developers, enabling complex arbitrage, collateral swap, and liquidation strategies.

- **Broad Asset Support:** The protocol supports a much wider and more diverse range of assets compared to more conservative platforms, including more volatile and niche tokens. This provides users with greater flexibility for collateral and borrowing options.

REFERENCES

[1] Aave Protocol. 2025. Aave Documentation. https://docs.aave.com/faq/ Accessed: 2025-10-01.

[2] Robert Leshner and Geoffrey Hayes. 2019. Compound: The Money Market Protocol. https://compound.finance/documents/Compound.Whitepaper.pdf Accessed: 2025-10-01.

[3] Sergey Nazarov, Steve Ellis, Ari Juels, et al. 2017. Chainlink: A Decentralized Oracle Network. https://chain.link/whitepaper Accessed: 2025-10-01.

[4] Uniswap Labs. 2020. Uniswap v2: Oracle. https://docs.uniswap.org/contracts/v2/concepts/advanced-topics/oracles Accessed: 2025-10-01.