# Privacy Preserving Decentralized Swap Derivative Leveraging Blockchain Technology and Cryptographic Primitives

Gayathri V, Kunwar Singh, and Karthik Narayan V V*

National Institute of Technology, Tiruchirappalli, TamilNadu, India
{gayathrivijayakumarme, kunwar2081, karthiknarayan0715}@gmail.com

**Abstract**

Blockchain technology is currently reshaping traditional business models by obviating the trusted third party. One specific aspect of Blockchain technology that has captivated the interest of numerous financial enterprises is Smart Contracts. These self-executing contracts have lot of potential in the financial industry. Eskandari et al. [1] have designed decentralized markets for option derivatives utilizing Ethereum blockchain technology. To the best of our knowledge there doesn't exists Blockchain based Interest Rate Swap. We are the first to propose a decentralized Interest Rate Swap derivative (IRS) using Ethereum Blockchain Technology. Furthermore, we have put forth a proposal for a privacy-preserving decentralized Interest Rate Swap derivative, leveraging both Ethereum Blockchain technology and Zether[2].

**Keywords:** Derivative, Interest Rate Swap, Smart contracts

## 1 Introduction

In 2007, financial crisis erupted in USA and many financial institutions (Trusted party) failed and bailed out by USA government with taxpayers' money. In this background, in 2008 Satoshi Nakamoto invented Bitcoin [3] in which individuals can make direct financial transaction without any trusted third party. Bitcoin was the first application of Blockchain technology. In Blockchain 2.0, smart contracts were deployed on Blockchain. A smart contract is executable code that runs on the Blockchain to facilitate, execute and enforce an agreement between untrusted parties without the involvement of a trusted third party. Although smart contract concept was introduced by Szabo in 1994 but there was no realtime implementation of it. Only in 2008 when the Bitcoin was invented by Satoshi Nakamoto, real implementation of smart contract was made possible. Smart contracts can automate complex contractual obligations without human interaction and can be part of business 4.0. Blockchain based smart contracts eliminates the intermediaries such as trusted third party. Thus it reduces the processing cost, improves the processing efficiency and makes real time transaction possible. There is need for Blockchain based smart contract in financial sector which will remove the middle man and improve the efficiency.

A Derivative involves two parties agreeing to a future transaction. Its value derives from (or depends on) the values of other underlying variable. Most common derivatives are Options, Forwards, Futures and Swaps. Eskandari et al. in [1], proposed and deployed decentralized market derivative and the given name as velocity. Velocity smart contract is implemented

The 7th International Conference on Mobile Internet Security (MobiSec'23), Dec. 19-21, 2023, Okinawa, Japan, Article No.44
*Corresponding author: National Institute of Technology, Trichy, Tiruchirappalli, Tamil Nadu 620015, India, Tel: +91-431-250-4135

for the option derivative in which parties buy and sell the options without the third party. Alfonso et al. in [4], proposed and designed the smart contract for forward derivatives. In [2], Benedikt Bunz et al. proposed Zether Smart Contract (ZSC), which provides privacy to the smart contract. Interest Rate Swap (IRS) is one of the swap derivatives, where interest rates are swapped between two parties once the swapping period is attained.

**Our Contribution:** Our contribution is twofold: first, we have proposed decentralized Interest Rate Swap derivative (IRS) using Ethereum Blockchain Technology. Second, we have proposed privacy preserving decentralized Interest Rate Swap derivative leveraging Ethereum Blockchain technology and Zether [2].

# 2 Preliminaries

## 2.1 Smart Contracts

A Contract is an agreement between two or more parties, which specifies certain enforceable things legally. In Smart Contract terms and conditions are written in programming language, and it is executed by the computer at any time. Smart Contracts are written in solidity language which works well on Ethereum Blockchain. Now a days Blockchain technology is used widely in many applications, because of its transparency and immutability. One of the most prominent fields where the Blockchain technology used is Financial Derivative. The input and output of the smart contract can be money or data.

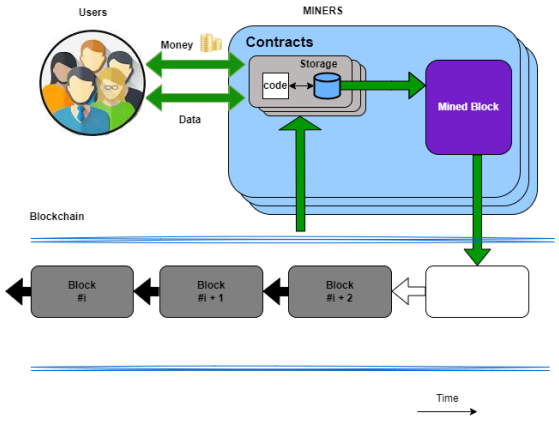


Figure 1: Working of smart contract[5]

## 2.2 Financial Derivative

A Derivative involves two parties agreeing to a future transaction. Its value derives from (or depends on) the values of other underlying variable. For example a stock option is a derivative whose value is dependent on the price of the stock. Most common derivatives are Options, Forwards, Futures and Swaps. Derivatives are used for hedging or speculating or arbitraging the risks. They may transfer wide range of risks from one entity to another in the economy [6].

## 2.3 Swaps

In India,the first swap deal dates back to 1984 when Oil and Natural Gas Corporation (ONGC) initiated the practice. However, the concept remained relatively inactive for quite some time. It wasn't until the mid-1990s, influenced by ONGC's pioneering move, that swaps were introduced

in the Indian markets by the Reserve Bank of India (RBI). Swap derivatives share similarities with forward contracts. A swap is an over-the-counter agreement between two companies to exchange cash flows in the future.[6]. The agreement defines the dates when the cash flows are to be paid and the way in which they are to be calculated. In essence, a forward contract represents the exchange of cash flows on a specific future date, while swaps involve the exchange of multiple cash flows on various future dates. From a trading perspective, swaps are negotiated through Over-the-Counter Agreements (OTC) between two companies, financial institutions, or individuals. These agreements outline the dates on which cash flows will be exchanged and the method for calculating them. Certain factors should be considered when trading swap derivatives.
Swap length: indicates the start and end of the contract
Swap term: indicates terms and conditions of the swap
Various types of swaps are available. The most commonly used swaps are Interest Rate Swap (IRS) and currency swap.

## 2.4 Interest Rate Swap

An Interest Rate Swap (IRS) is a swap derivative where interest rate at a predetermined fixed rate, applicable to a certain principal amount, is swapped for an interest at floating reference rate applied to the same principal amount, in which regular exchanges are happened for n agreed time period. Interest Rate Swap (IRS) is one of the swap derivatives, where interest rates are swapped between two parties once the swapping period is attained. Swapping period may be half-yearly or annually based upon the parties. The most commonly traded interest rate swaps are known as "vanilla" swaps, which swaps fixed-rate payments for floating-rate payments based on the reference rate. Commonly, reference rates are determined by some organization in every country. In Europe, it is determined by LIBOR (London Inter-Bank Offered Rate). LIBOR rates are the borrowing rates estimated by the banks in the inter bank market for periods between one day and one year. In IRS, the first party is called fixed rate payer and second party is called floating rate payer. Once the swapping period is attained, both parties' interest rates are swapped and they will receive their interest based on the new interest rates till the next swapping period. Till the end of the swapping agreement, party A agrees to receive Party B's interest amount in a certain time interval until the next swapping agreement period and vice versa. Notional principal is same for both parties. Only interest rates are swapped. Normally, swap derivatives are traded over-the-counter market. From Oracalize, smart contract retrieves the value of LIBOR and the Floating interest is calculated as follows.
Floating Interest Rate = LIBOR + Spread, where spread is the marginal value added to the interest rate and it is remains constant. With this floating interest, interest amount is calculated quarterly. One may refer [6] for more details about derivatives.

## 2.5 Over The Counter Market (OTC)

An Over-the-counter (OTC) market is a decentralized financial marketplace where participants engage in direct trading of various financial instruments, such as stocks, commodities, currencies, and more, without the involvement of a central exchange or intermediary broker. These markets operate electronically, lacking physical trading floors. OTC markets are primarily utilized for the trading of bonds, currencies, derivatives, and structured financial products.

### 2.5.1 Working of OTC

Over-the-counter (OTC) market encompasses the trading of securities outside the realm of major exchanges. This market hosts a diverse array of over 12,000 securities, including stocks, exchange-traded funds (ETFs), bonds, commodities, and derivatives. In contrast to traditional

exchanges like the New York Stock Exchange (NYSE) or Nasdaq, the OTC market lacks a physical trading venue. Instead, all transactions are executed electronically, directly between two parties, within a decentralized framework. Even though OTC securities are not officially "listed" on major exchanges, companies can still offer their stocks to the public through OTC channels. For the average investor, purchasing stocks in the OTC market may seem indistinguishable from acquiring exchange-listed securities. These OTC stocks are assigned unique ticker symbols and are typically available for trading via prominent online brokerage platforms. The OTC market serves as the default platform for certain securities, particularly corporate bonds. OTC derivatives are private contracts negotiated directly between counter parties without involving an exchange, although a broker might assist in the process. Brokers serve as market makers, setting stock prices, and transactions occur through brokers, not through formal exchanges like the New York Stock Exchange (NYSE).

### 2.5.2 Risks of Over-the-Counter Markets

Though swap derivatives have advantages, it has the following major risks.

1. Counter party Risk: In OTC markets, traders face substantial exposure to the possibility of their counterparts failing to meet their obligations due to the absence of clearinghouse. so, the parties are not able to make the current and future payments as per by the contract.

2. Lack of Transparency: Compared to the exchange-traded market, the OTC market typically offers reduced transparency. This is primarily due to the absence of centralized platforms where market participants can readily obtain information concerning trades, volumes, and pricing.

3. Regulatory Risk: OTC markets, in comparison to exchange-traded markets, often operate with lower levels of regulation which results in limited public information, possibility of outdated data, and potential for fraud.

## 2.6 Oracalize:

Oraclize, now known as "Provable", is the premier oracle service for smart contracts and blockchain applications, catering to numerous daily requests across platforms like Ethereum, Rootstock, R3 Corda, Hyperledger Fabric, and EOS. In essence, it serves as bridging the gap between the blockchain and the external real-world environment. It is a service designed to retrieve real-world data for smart contracts on the Ethereum blockchain. In the realm of blockchain technology, an oracle plays a vital role by supplying necessary data. This demand arises because blockchain applications, including Bitcoin scripts and smart contracts, lack the capability to directly access essential information. It acts as a crucial intermediary, supplying oracles that allow smart contracts to interact with external data sources, web APIs, and external systems. Oraclize plays a trusted role in the process by responsibly obtaining data from specified sources and securely delivering it to smart contracts. With oraclize, smart contracts can request a wide range of information from the Internet, including stock prices, sports scores, weather updates, accessing flight details and various other data types.

To use ProvableAPI, we have to first download the appropriate version from the following Github link, https://github.com/provable-things/ethereum-api/. First, we must make sure our smart contract inherits from the usingprovable contract present in the provableApi.sol file. Now we can use the provablequery function in the usingProvable contract to get the LIBOR rates by using web APIs. After getting the LIBOR rates, they can be used for variable interest calculation.

### 2.7 Zero knowledge Proof (ZKP)

In 1985, Shafi Goldwasser, Silvio Micali and Charles Rackoff, in their seminal paper introduced Zero-Knowledge Proofs (ZKP) [7]. In ZKP, one party (called prover) proves that a statement is true to another party (called verifier) without revealing anything except the fact that the statement is true. A Zero-Knowledge Proofs of Knowledge (ZKPoK) is a special case of ZKP when the statement consists only of the fact that the prover possesses the secret information. Later in 1988, Blum, Feldman and Micali proposed Non-Interactive Zero-Knowledge proofs (NIZK) [8]. In NIZK, the prover outputs just one message (called proof), which convinces the verifier the validity of the statement without revealing anything. A Zero-Knowledge protocol can be efficiently converted into a non-interactive zero-knowledge proof of knowledge in the random oracle model through the Fiat-Shamir transform [9].

## 3 Methodology

In the current OTC market, derivatives are traded between counter parties via trusted third parties for making contracts and exchanging money. In our paper, we have introduced an IRS derivative system without relying on a trusted third party. The functions of these intermediaries are performed by smart contracts, streamlining the process. Our contribution is twofold: First, We have proposed decentralized IRS (Version 01) using Ethereum Blockchain Technology. Second, We have also proposed privacy preserving decentralized IRS (Version 02) leveraging Ethereum Blockchain Technology and Zether[1].

### 3.1 Decentralized Interest Rate Swap (version-01)

We have developed and implemented a smart contract to execute an IRS that delivers transparency, mitigates counter party risk, and enhances security. Interest Rate Swap (IRS) is one of the swap derivatives, where interest rates are swapped between two parties once the swapping period is attained. We illustrate this through an example. Consider a scenario, Alice has deposited 1000 ETH in a Bank ABC, and agreed to receive floating interest rate quarterly till 1 year as swapping period. Alice will be getting interest four times in a year based on floating interest rate. The smart contract retrieves the floating interest rate from oraclize and computes the floating interest. Now Bob, who owns the same principal amount 1000 ETH in another Bank XYZ and agrees to receive fixed interest rate quarterly till 1 year as swapping period. After the swapping period is attained, Alice will be getting interest quarterly till 1 year based on fixed interest rate and Bob will be getting interest quarterly till 1 year based on floating interest rate. Swapping will continue alternatively till the maturity period.


Figure 2: Decentralisation of IRS version-01 Before Swapping

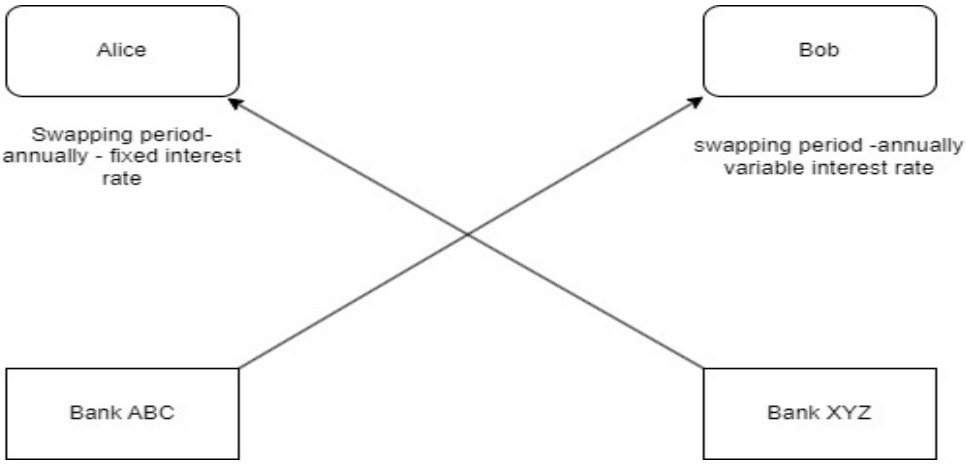


Figure 3: Decentralisation of IRS version-01 After Swapping

### 3.1.1 Implementation of Decentralized Interest Rate Swap (version-01):

Every quarter, the main program calls the claimInterests function present within the smart contracts. This function implementation is shown below.

```
function claimInterests() public onlyAliceOrBob(){

    //CHECKS IF CURRENT_TIME - LAST_SWAP_TIME > SWAP_INTERVAL
    if(block.timestamp - lastInterestSwapTime > SWAP_INTERVAL){
        lastInterestSwapTime = block.timestamp;
        swapInterests();
    }

    uint256 alice_interest = claimAliceInterest();
    uint256 bob_interest = claimBobInterest();

    bob.interestAmount += bob_interest;
    alice.interestAmount += alice_interest;
    bob.balance += bob_interest;
    alice.balance += alice_interest;

    emit ValueReturned(alice_interest, bob_interest);
}
```

Figure 4: Version 1: Function to claim interests and handle swaps

Upon invocation, claimInterests() function verifies if the time elapsed since the last swap exceeds the SWAP_INTERVAL. If this condition holds true, it toggles the interest rate calculation method between fixed and variable. Next alice_interest and bob_interest are fetched by calling the claimAliceInterest and claimBobInterest functions respectively.

```
// Function for Alice to claim her interest
function claimAliceInterest() public returns (uint256) {
    require(block.timestamp < contractStartTime + alice.maturityPeriodInSeconds, "Maturity period has ended");

    // Calculate Alice's interest since the last update
    uint256 alice_interest;
    if(alice.variableInterest){
        alice_interest = calculateVariableInterest(alice);
    }
    else{
        alice_interest = calculateFixedInterest(alice);
    }
    alice.lastInterestUpdateTime = block.timestamp;

    return alice_interest;
}
```

Figure 5: Version 1: Function to claim Alice interests

This function is dedicated to compute Alice's interest using her specified interest rate. A similar function, claimBobInterest, has been implemented for calculating Bob's interest. We calculate fixed interest and variable interest by using the functions calculateFixedInterest and calculateVariableInterest respectively. These functions compute interest according to the chosen interest rate. In the case of variable interest calculation, the provable_query function is employed to retrieve variable interest rates from external web APIs, such as LIBOR rates.

```
function calculateFixedInterest(Account memory account) view internal returns (uint256){
    uint256 timeElapsed = (block.timestamp - account.lastInterestUpdateTime) / (1 seconds);
    uint256 computedInterest;
    uint256 computedInterest = account.principalAmount * account.interest_rate;
    computedInterest *= timeElapsed;
    computedInterest /= 100;

    return computedInterest;
}

function calculateVariableInterest(Account memory account) internal returns (uint256){
    uint256 timeElapsed = (block.timestamp - account.lastInterestUpdateTime) / (1 seconds);
    uint256 liborRate = uint256(provable_query("URL", "json(URL_TO_QUERY)"));
    uint256 computedInterest = alice.principalAmount * liborRate;
    computedInterest *= timeElapsed;
    computedInterest /= 100;
    return computedInterest;
}
```

Figure 6: Version 1: Functions to calculate fixed and variable interests

## 3.2 Privacy Preserving Decentralized IRS (Version-02):

In IRS version 01, Blockchain reveals the amount of money deposited by Alice and Bob. Also it reveals the interest given to both parties. We have developed another smart contract to execute an IRS which provides privacy leveraging Zether[1].
Public Parameter (n): On input of security parameter $n$, two primes p and q are chosen such that p-1=2q. Let Zp be a cyclic group with generator 'h'. Create a group that is generated from $h^2$=g. Let G be a cyclic subgroup of Zp and it is called a Schnorr group with generator g. The public parameter of Schnorr group G with generator g.
Key Generation (n): Alice and Bank choose randomly '$x_a$' and '$x$' $\in$ 2,...,q-1 and Computes their public key $g^{x_a}$ and $g^x$
$y_a = g^{x_a}, y = g^x$
Private key (SK) of Alice = $(x_a)$
Public Key (PK) of Alice = $(y_a)$
Private key (SK) of Bank =$(x)$
Public Key (PK) of Bank = $(y)$
**Step 1:** Alice has deposited 'b' ether by sending the cipher text $(C_{1a}, C_{2a})$ to IRS Smart Contract. This Cipher text $(C_{1a}, C_{2a})$ is computed by encrypting the 'b' ether by using bank's public key $y$ where, $(C_{1a}, C_{2a}) = (g^b\ y^r, g^r\ ) = (g^b, 1) = C_a$ (randomness r =0 )
she also sends Zero Knowledge Proof(ZKP) that b is greater than zero using Bulletproof b >0 [10].
**Step 2:** Bank ABC has deposited $b^{'}$ ether by sending the cipher text $(C_{1B}, C_{2B})$ to IRS Smart Contract. This Cipher text $(C_{1B}, C_{2B})$ is computed by encrypting the $b^{'}$ ether by using bank's public key $y$ where, $(C_{1B}, C_{2B}) = (g^{b^{'}}\ y^{r1}, g^{r1})$ (randomness $r_1$ )
$C_B$=$(g^{b^{'}}\ y^{r1}, g^{r1})$. Bank also sends Zero Knowledge Proof(ZKP) that $b^{'}$ is greater than '0' and 'b' $b^{'}$ >0 and $b^{'}$>b [10].

**Step 3:** Now the balance in the IRS smart contract is computed by multiplying these cipher texts as follows. $(C_1, C_2)$ = $(C_{1a}, C_{2a})$ * $(c_{1B}, C_{2B})$ = ($g^b$,1) * ($g^{b'} y^{r1}, g^{r1}$)
$C$=($g^{b+b'} y^{r1}, g^{r1}$)
**Step 4:** After one quarter, smart contract calculates the encrypted interest using Alice's public key as follows.
As we know, interest $b^*$=b * I, where I=$i_1$ / (100 * 4)
Now the smart contract computes,
$g^{b^*}$ as $(g^b)^I$
Now the encrypted Interest = ($C_1^*$,$C_2^*$)
$C^*$=($C_1^*$,$C_2^*$)=($g^{b.I} y_a^{r2}, g^{r2}$) (randomness r2)
**Step 5:** Now this interest amount $b^*$ has to be transferred to Alice's account. This encrypted interest is sent to Alice's account and the same is subtracted from IRS smart contract. Now the balance in smart contract is computed by dividing the $C/C^*$
$C_{bal}$=($C_1/C_1^*$), ($C_2/C_2^*$)
($C_{1bal}$,$C_{2bal}$)=($g^{b+b'} y^{r1}$,$g^{r1}$) / ($g^{b.I} y_a^{r2}, g^{r2}$)
($C_{1bal}$,$C_{2bal}$)=($g^{b+b'-b-I}.y^{r1-r2}, g^{r1-r2}$)
$C_{bal}$=($C_{1bal}, C_{2bal}$)=($g^{b'-I}.y^{r1-r2}, g^{r1-r2}$)
**Step 6:** since Alice knows the amount $b^*$, she has some guess about $g^{b^*}$. so, she can can compute $g^{b^*}$ in polynomial time.
**Step 7:** Alice can burn this encrypted amount $g^b$ into real ether by giving ZKP that she knows 'b' ether.This ZKP is given in the Appendix.
**Step 8:** Similar process is done by Bob and another Bank XYZ.
**Step 9:** Once the swapping period is attained,interest is swapped and the same process is repeated till the maturity period.
**Burn Transaction:** Alice gets the encrypted amount as $(C_L, C_R)$ = ($g^b$ $y_a^r$, $g^r$). She has to convert this encrypted amount to $b$ ether. Alice decrypts this cipher text and gets $g^b$. Since she knows the amount $b$ and can verify with decrypted amount $g^b$. Now she has to prove that she knows her private key $x_a$, where $y_a = g^{x_a}$ is her public key and ($g^b$ $y_a^r$, $g^r$) is valid encryption of $b$ under $y_a$. We can write $C_L$ =$C_R{}^{x_a}$. Now she has to prove that she knows secret key $x_a$ in following statement:
$y = g^{sk} h^0 \Lambda C_L = y^b C_R^{sk}$
We have written non-interactive zero knowledge proof for the knowledge of secret key $x_a$ as follows which is based on [11].
**Step 1:** Prover generates 3 random numbers $k_1$,$k_2$,$k_3$ and and creates 2 commitments
$t_1$=$g^{k_1}$ * $h^{k_2}$ , $t_2 = g^{r*k_1} * y^{k_3}$
**Step 2:** Prover creates challenge c = hash ($g$,$g^r$,$h$,$y$,p,q,$t_1$,$t_2$)
**Step 3:** Prover creates responses as $s_1$=$k_1$ + ($x_a$ *c)
$s_2$=$k_2$ + (0*c)
$s_3$=$k_3$ + (b*c) and sends ($t_1$,$t_2$,$s_1$,$s_2$,$s_3$) to the verifier
**Step 4:** Verifier checks if $g^{s_1}$ * $h^{s_2} = p^c * t_1$ and $g^{r*s_1} * y^{s_3} = q^c * t_2$ If it is equal Alice will be able to claim the amount.

### 3.2.1 Implementation of Privacy Preserving Decentralized IRS (Version-02)

The implementation of version-02 is very similar to version 1. The main change with version-02 is that the principal amount, and interest rates are encrypted using the public key. The principal amount $b$ is encrypted and it is stored as $g^b$. For the smart contract version-02, the input is $g^b$ and the interest is calculated as $(g^b)^I$. The balance amount is stored $g^{b'}$. This gives

us an added layer of security.The functions claimAliceInterest and claimBobInterest remains same for both the version.

```
function claimInterests() public onlyAliceOrBob(){

    //CHECKS IF CURRENT_TIME - LAST_SWAP_TIME > SWAP_INTERVAL
    if(block.timestamp - lastInterestSwapTime > SWAP_INTERVAL){
        lastInterestSwapTime = block.timestamp;
        swapInterests();
    }

    uint256 alice_interest = claimAliceInterest();
    uint256 bob_interest = claimBobInterest();

    bob.interestAmount *= bob_interest;
    alice.interestAmount *= alice_interest;
    bob.balance *= bob_interest;
    alice.balance *= alice_interest;
    emit ValueReturned(alice_interest, bob_interest);
}
```

Figure 7: Version 2:Function to claim interests and handle swaps

```
function calculateFixedInterest(Account memory account) internal view returns (uint256){
    uint256 timeElapsed = (block.timestamp - account.lastInterestUpdateTime) / (1 seconds);
    uint256 computedInterest;
    uint256 power = account.interest_rate;
    power *= timeElapsed;
    power /= 100;
    computedInterest = (account.principalAmount ** (power));

    return computedInterest;
}
function calculateVariableInterest(Account memory account) internal returns (uint256){
    uint256 timeElapsed = (block.timestamp - account.lastInterestUpdateTime) / (1 seconds);
    uint256 computedInterest;
    uint256 liborRate = uint256(provable_query("URL", "json(URL_TO_QUERY).data"));
    uint256 power = liborRate;
    power *= timeElapsed;
    power /= 100;
    computedInterest = (account.principalAmount ** (power));
    return computedInterest;
}
```

Figure 8: Version 2: Functions to calculate fixed and variable interests

### 3.2.2 Relating to Mobile security

Let Alice and Bob has deposited 1000 ETH in a bank ABC and bank XYZ via mobile and agreed to receive floating interest and fixed interest quarterly and 1 year as swapping period till maturity period respectively. So, by using our both versions of IRS, a user can perform the swap derivative in safe and secure manner using mobile phone. This way our paper is related to MobiSec 2023.

### 3.2.3 Result

By implementing the IRS derivative in a decentralized way, we have achieved the following results. Comparative study is made between the traditional OTC and our proposed decentral-

ization methods (both versions) and the following parameters are analysed.

| S.No | Parameter | OTC | Our Work | |
|---|---|---|---|---|
| | | | IRS Version 1 | IRS Version 2 |
| 1. | Centralized or decentralized | decentralized | decentralized | decentralized |
| 2. | Transparency | no | yes | no |
| 3. | counter party | yes | no | no |
| 4. | Regulatory risk | yes | no | no |
| 5. | Privacy | no | no | yes |

# 4 Conclusion

Derivatives are the key area for automation through smart contracts. In our research, we have designed and deployed blockchain based swap derivatives. We examined how swap derivatives are traded in the OTC market and identified issues like counter party risk, regulatory risk, and lack of transparency. In first version, we have designed and developed framework for decentralized IRS derivative. In second version we have added the privacy to the IRS derivative.

# 5 Acknowledgements

This research is undertaken as part of the project 'Research and Development of Secure and Privacy Preserving Blockchain based Smart Contract and its Applications' funded by Science and Engineering Research Board (SERB) [EEQ/2021/000305].

# References

[1] Shayan Eskandari, Jeremy Clark, Vignesh Sundaresan, and Moe Adham. On the feasibility of decentralized derivatives markets. In Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson, editors, *Financial Cryptography and Data Security*, pages 553–567, Cham, 2017. Springer International Publishing.

[2] Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. Zether: Towards privacy in a smart contract world. In Joseph Bonneau and Nadia Heninger, editors, *Financial Cryptography and Data Security*, pages 423–443, Cham, 2020. Springer International Publishing.

[3] Satoshi Nakamoto and A Bitcoin. A peer-to-peer electronic cash system. *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf*, 4(2):15, 2008.

[4] Alfonso D. D. M. Rius and Eamonn Gashier. Smart derivatives: On-chain forwards for digital assets. In Tiziana Margaria and Bernhard Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation: Applications*, pages 195–211, Cham, 2020. Springer International Publishing.

[5] Peter Eze, Tochukwu Eziokwu, and Chinedu Okpara. A triplicate smart contract model using blockchain technology. *Circulation in Computer Science*, DC CPS 2017:1–10, 06 2017.

[6] John C Hull. *Options futures and other derivatives*. Pearson Education India, 2003.

[7] Shafi Goldwasser, Silvio Micali, and Chales Rackoff. *The Knowledge Complexity of Interactive Proof-Systems*, page 203–225. Association for Computing Machinery, New York, NY, USA, 2019.

[8] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge proof systems. In *Advances in Cryptology—CRYPTO'87: Proceedings 7*, pages 52–72. Springer, 1988.

[9] Charles Rackoff and Daniel R Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Annual international cryptology conference*, pages 433–444. Springer, 1991.

[10] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334, 2018.

[11] Lovesh Harchadani. Zero knowledge proofs with sigma protocols, 2019. June 16, 2019.