Central bank digital currency
Threats and vulnerabilities

Defcon 29

Ian Vitek
CBDC Security

Central bank of Sweden, Sveriges Riksbank
Central bank digital currency
Threats and vulnerabilities

Presentation

Background

Detailed system description of the prototype

Vulnerabilities in the retail central bank digital currency prototype

Everything else

Solutions and summary
So where do we start?

Ian Vitek

• Started with pentests 1996.
• Interested in web application security, network layer 2 (the writer of macof), DMA attacks and local pin bypass attacks (found some on iPhone).

Sveriges Riksbank (Central bank of Sweden)

Disclaimer: The views and opinions expressed in this presentation are those of the presenter and do not necessarily represent the views and opinions of the Riksbank.
The e-krona project

• Why central bank digital currency?
• Procurement
  • Requirements
  • Winning bid
• Work on the prototype phase 1 (year one)

The goal of this presentation is to share insights of the security challenges of building a prototype of a two tier retail central bank digital currency.
Detailed system description of the prototype
Detailed system description of the prototype

User
Detailed system description of the prototype

Security and logic, e.g.
- PIN
- Signing transactions
- Encryption
Detailed system description of the prototype

Security and logic, e.g.
- PIN
- Signing transactions
- Encryption

Storage of
- PIN
- Private key for tokens
- Authentication keys
- Message keys, e.g. Firebase
Detailed system description of the prototype

Security and logic, e.g.
- PIN
- Signing transactions
- Encryption

Storage of
- PIN
- Private key for tokens
- Authentication keys
- Message keys, e.g. Firebase
Detailed system description of the prototype

Security and logic, e.g.
- Authentication
- Push messages, e.g. Firebase
- Limits
- Back office functions

App

PSP1 business logic

Disk
Detailed system description of the prototype

Security and logic, e.g.
- Authentication
- Push messages, e.g. Firebase
- Limits
- Back office functions

Storage of
- Authentication keys
- Payment history
- Customer data
- Message keys, e.g. Firebase

Diagram:
- App
- Disk
- DB
- PSP1 business logic

Diagram arrows indicate interaction between App, Disk, DB, and PSP1 business logic.
Detailed system description of the prototype

- App
- Disk
- PSP1 business logic
- PSP1 Corda node

Security and logic, e.g.
- Token transactions
- Token verification
- Wallet management
Detailed system description of the prototype

- Security and logic, e.g.
  - Token transactions
  - Token verification
  - Wallet management

- Storage of
  - Public keys (and PSP1 private keys)
  - Wallets
  - Tokens
  - Backchain
  - Certificates
Detailed system description of the prototype

Security and logic, e.g.
- Issue and redeem
- Token verification
- Corda network management
Detailed system description of the prototype

Security and logic, e.g.
- Issue and redeem
- Token verification
- Corda network management

Storage of
- Public keys (and Riksbank private keys)
- Backchain
- Certificates
Detailed system description of the prototype

Security and logic, e.g.
- Prevent double-spends
- Sign the transaction
Detailed system description of the prototype

Security and logic, e.g.
- Prevent double-spends
- Sign the transaction

Storage of
- Notary private key
- Hashes of tokens
- Certificates
Detailed system description of the prototype

Security and logic, e.g.
• Interest
• Back office functions
Detailed system description of the prototype

Security and logic, e.g.
- Interest
- Back office functions

Storage of
- Outstanding CBDC
Detailed system description of the prototype
Detailed system description of the prototype

Security and logic, e.g.
- Add and remove alias
- Map alias to PSP and wallet

- [Diagram of system components]
Security and logic, e.g.
- Add and remove alias
- Map alias to PSP and wallet

Storage of
- Alias ⇔ PSP and wallet

Detailed system description of the prototype
Detailed system description of the prototype
What is backchain?
And how to exploit bad implementation...
What is backchain?
And how to exploit bad implementation...
What is backchain?
And how to exploit bad implementation...

Riksbank
Corda node

Issue
1000

Corda node

PSP1

node
What is backchain?
And how to exploit bad implementation...

Transactions
And tokens
What is backchain?
And how to exploit bad implementation...

Riksbank
Corda
node

Issue
1000
PSP1
Corda
node

Transactions
And tokens
What is backchain?
And how to exploit bad implementation...

Transactions
And tokens

Transactions
And tokens

Riksbank
Corda
node

Issue
1000

PSP1
Corda
node

Tx: 1
What is backchain?
And how to exploit bad implementation...

- Riksbank Corda node
- Issue 1000
- PSP1 Corda node

Transactions and tokens
- Token#: 1[0]
- Amount: 1000
- Owner: PSP1
- Sign: Riksbank
- Reference: None
What is backchain?
And how to exploit bad implementation...

Riksbank Corda node → Issue 1000 → PSP1 Corda node → UserA withdraw 200 → PSP1 Corda node

Transactions And tokens → Tx: 1

Token#: 1[0]
Amount: 1000
Owner: PSP1
Sign: Riksbank
Reference: None
What is backchain?
And how to exploit bad implementation...

Transactions And tokens

<table>
<thead>
<tr>
<th>Riksbank Corda node</th>
<th>Issue 1000</th>
<th>PSP1 Corda node</th>
<th>UserA withdraw 200</th>
<th>PSP1 Corda node</th>
</tr>
</thead>
</table>

Token#: 1[0]
Amount: 1000
Owner: PSP1
Sign: Riksbank
Reference: None
What is backchain?
And how to exploit bad implementation...

Transactions
And tokens

Tx: 1

Issue

1000

Riksbank
Corda node

PSP1
Corda node

UserA withdraw

200

PSP1
Corda node

Token#: 1[0]
Amount: 1000
Owner: PSP1
Sign: Riksbank
Reference: None

Tx: 2
What is backchain?
And how to exploit bad implementation...

Transactions
And tokens

Riksbank
Corda
node

Issue
1000

PSP1
Corda
node

UserA withdraw
200

PSP1
Corda
node

Token#: 1[0]
Amount: 1000
Owner: PSP1
Sign: Riksbank
Reference: None

Tx: 1

Token#: 2[0]
Amount: 200
Owner: UserA
Sign: PSP1
Reference: 1[0]

Tx: 2
What is backchain?
And how to exploit bad implementation...

Riksbank
Corda node

PSP1
Corda node

UserA withdraw

1000

200

PSP1
Corda node

Issue

Token#:
1[0]
Amount:
1000
Owner:
PSP1
Sign:
Riksbank
Reference:
None

Token#:
2[0]
Amount:
200
Owner:
UserA
Sign:
PSP1
Reference:
1[0]

35
What is backchain?
And how to exploit bad implementation...

Transactions

And tokens

<table>
<thead>
<tr>
<th>Riksbank Corda node</th>
<th>Issue 1000</th>
<th>PSP1 Corda node</th>
<th>UserA withdraw 200</th>
<th>PSP1 Corda node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token#: 1[0]</td>
<td>Amount: 1000</td>
<td>Owner: PSP1</td>
<td>Sign: Riksbank</td>
<td>Reference: None</td>
</tr>
<tr>
<td>Token#: 2[0]</td>
<td>Amount: 200</td>
<td>Owner: UserA</td>
<td>Sign: PSP1</td>
<td>Reference: 1[0]</td>
</tr>
<tr>
<td>Token#: 2[1]</td>
<td>Amount: 800</td>
<td>Owner: PSP1</td>
<td>Sign: PSP1</td>
<td>Reference: 1[0]</td>
</tr>
</tbody>
</table>
What is backchain?
And how to exploit bad implementation...

- **Riksbank Corda node**
  - Issue: 1000
- **PSP1 Corda node**
  - UserA withdraw: 200
- **PSP1 Corda node**
  - UserA => UserB: 50

Transactions and tokens:

1. **Token**: 1[0]
   - Amount: 1000
   - Owner: PSP1
   - Sign: Riksbank
   - Reference: None
   - **Tx**: 1

2. **Token**: 2[0]
   - Amount: 200
   - Owner: UserA
   - Sign: PSP1
   - Reference: 1[0]
   - **Tx**: 2

3. **Token**: 2[1]
   - Amount: 800
   - Owner: PSP1
   - Sign: PSP1
   - Reference: 1[0]
What is backchain?
And how to exploit bad implementation...

Transactions And tokens

- Riksbank Corda node
  - Issue 1000
  - PSP1 Corda node
  - UserA withdraw 200
  - PSP1 Corda node
  - UserA => UserB 50
  - PSP2 Corda node

Transactions And tokens

- Token#: 1[0]
  - Amount: 1000
  - Owner: PSP1
  - Sign: Riksbank
  - Reference: None

- Token#: 2[0]
  - Amount: 200
  - Owner: UserA
  - Sign: PSP1
  - Reference: 1[0]

- Token#: 2[1]
  - Amount: 800
  - Owner: PSP1
  - Sign: PSP1
  - Reference: 1[0]
What is backchain?
And how to exploit bad implementation...

- Riksbank
  - Issue
  - PSP1
  - UserA withdraw
  - PSP1
  - UserA => UserB
  - PSP2

Transactions And tokens

- Token#: 1[0]
  - Amount: 1000
  - Owner: PSP1
  - Sign: Riksbank
  - Reference: None

- Token#: 2[0]
  - Amount: 200
  - Owner: UserA
  - Sign: PSP1
  - Reference: 1[0]

- Token#: 2[1]
  - Amount: 800
  - Owner: PSP1
  - Sign: PSP1
  - Reference: 1[0]
What is backchain?
And how to exploit bad implementation...

Riksbank Corda node
Issue 1000
PSP1 Corda node
UserA withdraw 200
PSP1 Corda node
UserA => UserB
PSP2 Corda node

Transactions And tokens

Token#: 1[0]
Amount: 1000
Owner: PSP1
Sign: Riksbank
Reference: None

Token#: 2[0]
Amount: 200
Owner: UserA
Sign: PSP1
Reference: 1[0]

Token#: 3[0]
Amount: 50
Owner: UserB
Sign: UserA
Reference: 2[0]

Token#: 2[1]
Amount: 800
Owner: PSP1
Sign: PSP1
Reference: 1[0]
What is backchain?
And how to exploit bad implementation...

Transactions

And tokens

Riksbank
Corda
node

Issue
1000

PSP1
Corda
node

UserA withdraw
200

PSP1
Corda
node

UserA => UserB
50

PSP2
Corda
node

Token#: 1[0]
Amount: 1000
Owner: PSP1
Sign: Riksbank
Reference: None

Token#: 2[0]
Amount: 200
Owner: UserA
Sign: PSP1
Reference: 1[0]

Token#: 3[0]
Amount: 50
Owner: UserB
Sign: UserA
Reference: 2[0]

Token#: 2[1]
Amount: 800
Owner: PSP1
Sign: PSP1
Reference: 1[0]

Token#: 3[1]
Amount: 150
Owner: UserA
Sign: UserA
Reference: 2[0]
What is backchain? And how to exploit bad implementation...

Riksbank Corda node Issue 1000 PSP1 Corda node UserA withdraw 200 PSP1 Corda node UserA => UserB 50 PSP2 Corda node

Transactions And tokens

Tx: 1
Token#: 1[0]
Amount: 1000
Owner: PSP1
Sign: Riksbank
Reference: None

Tx: 2
Token#: 2[0]
Amount: 200
Owner: UserA
Sign: PSP1
Reference: 1[0]

Tx: 3
Token#: 3[0]
Amount: 50
Owner: UserB
Sign: UserA
Reference: 2[0]

Token#: 2[1]
Amount: 800
Owner: PSP1
Sign: PSP1
Reference: 1[0]

Token#: 3[1]
Amount: 150
Owner: UserA
Sign: UserA
Reference: 2[0]
How to exploit token selection and long backchains

PSP1 UserA

Token#1
200
PSP1
How to exploit token selection and long backchains

PSP1 UserA

Withdrawal 3 and deposit 2

| Token#1 | 200 | PSP1 |
How to exploit token selection and long backchains

Withdrawal 3 and deposit 2

PSP1 UserA

Token#1
200
PSP1

Token#2
197
PSP1

Token#3
3
UserA
How to exploit token selection and long backchains

Withdrawal 3 and deposit 2

PSP1 UserA

Token#1
200 PSP1

Token#2
197 PSP1

Token#3
3 UserA

Token#5
1 UserA

Token#6
2 PSP1
How to exploit token selection and long backchains

Historic Transactions

Withdrawal 3 and deposit 2

PSP1 UserA

Token#1
200
PSP1

Token#2
197
PSP1

Token#3
3
UserA

Token#5
1
UserA

Token#6
2
PSP1
How to exploit token selection and long backchains

Historic Transactions

Withdrawal 3 and deposit 2

Token#1
200
PSP1

Token#2
197
PSP1

Token#3
3
UserA

Token#5
1
UserA

Token#6
2
PSP1
How to exploit token selection and long backchains

Historic Transactions

Withdrawal 3 and deposit 2

PSP1 UserA

Token#1
200
PSP1

Token#2
197
PSP1

Token#3
3
UserA

Token#5
1
UserA

Token#6
2
PSP1
How to exploit token selection and long backchains

So UserA does this over and over again

<table>
<thead>
<tr>
<th>Historic Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token#1</td>
</tr>
<tr>
<td>Token#3</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>UserA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSP1 UserA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token#2</td>
</tr>
<tr>
<td>197</td>
</tr>
<tr>
<td>PSP1</td>
</tr>
<tr>
<td>Token#5</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>UserA</td>
</tr>
<tr>
<td>Token#6</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>PSP1</td>
</tr>
</tbody>
</table>
How to exploit token selection and long backchains

So UserA does this over and over again
How to exploit token selection and long backchains

So UserA does this over and over again
How to exploit token selection and long backchains

So UserA does this over and over again

Historic Transactions

- Token#1
- Token#3
- Token#2
- Token#8
  - 3
  - PSP1

PSP1 UserA

- Token#7
  - 194
  - PSP1

- Token#5
  - 1
  - UserA

- Token#6
  - 2
  - PSP1

- Token#9
  - 1
  - UserA

- Token#10
  - 2
  - PSP1
How to exploit token selection and long backchains

Historic Transactions

<table>
<thead>
<tr>
<th>Token#1</th>
<th>Token#3</th>
<th>Token#2</th>
<th>Token#8</th>
<th>Token#x</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserA</td>
<td>UserA</td>
<td>UserA</td>
<td>UserA</td>
<td>UserA</td>
</tr>
<tr>
<td>PSP1</td>
<td>PSP1</td>
<td>PSP1</td>
<td>PSP1</td>
<td>PSP1</td>
</tr>
</tbody>
</table>

Many times

<table>
<thead>
<tr>
<th>Token#5</th>
<th>Token#9</th>
<th>Token#x</th>
<th>Token#x</th>
<th>Token#x</th>
<th>Token#x</th>
<th>Token#x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>UserA</td>
<td>UserA</td>
<td>UserA</td>
<td>UserA</td>
<td>UserA</td>
<td>UserA</td>
<td>UserA</td>
</tr>
<tr>
<td>PSP1</td>
<td>PSP1</td>
<td>PSP1</td>
<td>PSP1</td>
<td>PSP1</td>
<td>PSP1</td>
<td>PSP1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Token#x</th>
<th>44</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSP1</td>
<td></td>
</tr>
</tbody>
</table>

PSP1 UserA

<table>
<thead>
<tr>
<th>Token#x</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSP1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Token#x</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserA</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Token#x</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSP1</td>
<td></td>
</tr>
</tbody>
</table>
How to exploit token selection and long backchains

Historic Transactions

Deposit all tokens

PSP1 UserA

Token#1
Token#3
Token#2
Token#8
Token#x
Token#x
Token#x
Token#x
Token#x
Token#x
Token#x
Token#x
Token#x
PSP1

Token#5
Token#9
Token#x
Token#x
Token#x
Token#x
Token#x
Token#x
Token#x
Token#n

Token#1
UserA
Token#1
UserA
Token#1
UserA
Token#1
UserA
Token#1
UserA
Token#1
UserA

Token#2
PSP1
Token#2
PSP1
Token#2
PSP1
Token#2
PSP1
Token#2
PSP1
Token#2
PSP1

Token#3
PSP1
Token#3
PSP1
Token#3
PSP1
Token#3
PSP1
Token#3
PSP1
Token#3
PSP1

Token#x
44
PSP1
How to exploit token selection and long backchains

PSP1 Admin

Redeem

Riksbank Corda node
Other setups with better effects

Several issue tokens to get several merkle trees

1500 PSP1

2000 PSP1

3000 PSP1

100 UserA
Other setups with better effects

Several issue tokens to get several merkle trees

Split tokens into hundreds

1500 PSP1

2000 PSP1

3000 PSP1

100 UserA

1 UserB

1 UserB

1 UserB

1 UserB

1 UserB
Other setups with better effects

Several issue tokens to get several merkle trees

1500
PSP1

2000
PSP1

3000
PSP1

100
UserA

Split tokens into hundreds

1
UserB

1
UserB

1
UserB

1
UserB

UserA

Use hundreds of tokens in one transaction

5
UserA

59
Other setups with better effects

Several issue tokens to get several merkle trees

1500 PSP1

2000 PSP1

3000 PSP1

100 UserA

Split tokens into hundreds

1 UserB
1 UserB
1 UserB
1 UserB
1 UserB

Use hundreds of tokens in one transaction

5 UserA

And do this over and over again to get the TransactionOfDeath
Crash nodes with TransactionOfDeath and permanently lock tokens

Sometimes crashes gives inconsistencies.

Token#1
5
UserA

UserA => UserC

Token#2
5
UserC

PSP1
Corda node

PSP3
Corda node
Crash nodes with TransactionOfDeath and permanently lock tokens

Sometimes crashes give inconsistencies.

- **Token#1**
  - 5
  - UserA

- **UserA => UserC**

- **Token#2**
  - 5
  - UserC

- **PSP1**
  - Corda node

- **PSP3**
  - Corda node

Mark Token#1 as used

- **Riksbank**
  - Corda notary
Crash nodes with TransactionOfDeath and permanently lock tokens

Sometimes crashes gives inconsistencies.

Mark Token#1 as used

Riksbank Corda notary
Crash nodes with TransactionOfDeath and permanently lock tokens

Sometimes crashes give inconsistencies.

Available token: Token#1

Mark Token#1 as used

Used token: Token#1

Riksbank Corda notary

Token#1
5
UserA

UserA => UserC

Token#2
5
UserC

RSP1 Corda node

RSP3 Corda node

Token#1

Available token: Token#1

Mark Token#1 as used

Used token: Token#1

Riksbank Corda notary

Token#2
5
UserC
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.

UserC

PSP1
Corda node

UserC => UserA

PSP3
Corda node
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.

UserC

PSP1
corda
node

UserC => UserA

PSP3
corda
node

Token#2
5
UserA

Token#1
5
UserC
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.

Mark Token#1 as used
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.

Mark Token#1 as used

Riksbank Corda notary
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.
Network problems, timeouts, in memory token selection and lock tokens until restarted

Card payments in the prototype phase 1 is a signed transaction on the smart card traveling through the PSP of the Merchant to card holder PSP.

UserC

PSP1
Corda node

UserC => UserA

PSP3
Corda node

Token#2
5
UserA

Token#1
5
UserC

Timeout!

Error!

Mark Token#1 as used

Riksbank
Corda notary

Available token: Token#1

Used token: Token#1
Evil PSP can lock tokens of other PSPs

As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.

EvilPSP
Corda node

UserA => UserC

PSP3
Corda node

Evil PSP can lock tokens of other PSPs

As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.

EvilPSP  
Corda node  
Token#1  
5  
UserA  

UserA => UserC  
PSP3  
Corda node  
Token#2  
5  
UserC  

Evil PSP can lock tokens of other PSPs

As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.

![Diagram showing the flow of tokens between different entities.]

Riksbank Corda notary

Token#1
5
UserA

UserA => UserC

PSP3 Corda node

Token#2
5
UserC

Mark Token#1 as used

In:ing.com

Evil PSP can lock tokens of other PSPs

As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.

EvilPSP

Corda node

UserA => UserC

PSP3

Corda node

Token#1

5

UserA

Token#2

5

UserC

Mark Token#1 as used

Used token: Token#1

Available token: Token#2

Riksbank

Corda notary

ing.com

Evil PSP can lock tokens of other PSPs

As the evil PSP has the information about the tokens sent to others, the evil PSP can send those tokens to the non validating notary node to be marked as used.

![Diagram showing the process of Evil PSP locking tokens of other PSPs.](https://www.ingwb.com/media/3024436/solutions-for-the-corda-security-and-privacy-trade-off_-whitepaper.pdf)
And an ending note on token selection

PSP1 Corda

<table>
<thead>
<tr>
<th>Token#1</th>
<th>Token#2</th>
<th>Token#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000000</td>
<td>1000000</td>
<td>1000000</td>
</tr>
<tr>
<td>Owner: PSP1</td>
<td>Owner: PSP1</td>
<td>Owner: PSP1</td>
</tr>
</tbody>
</table>
And an ending note on token selection

PSP1 Corda

<table>
<thead>
<tr>
<th>Token#1</th>
<th>Token#2</th>
<th>Token#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000000</td>
<td>1000000</td>
<td>1000000</td>
</tr>
<tr>
<td>Owner: PSP1</td>
<td>Owner: PSP1</td>
<td>Owner: PSP1</td>
</tr>
</tbody>
</table>

PSP1 Admin  

PSP1 UserA

Redeem 3500000  
Withdraw 50
And an ending note on token selection

PSP1 Admin

PSP1 UserA

Redeem 3500000
Withdraw 50

<table>
<thead>
<tr>
<th>PSP1 Corda</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token#1</strong></td>
</tr>
<tr>
<td>2000000</td>
</tr>
<tr>
<td>Owner: PSP1</td>
</tr>
<tr>
<td><strong>Token#2</strong></td>
</tr>
<tr>
<td>1000000</td>
</tr>
<tr>
<td>Owner: PSP1</td>
</tr>
<tr>
<td><strong>Token#3</strong></td>
</tr>
<tr>
<td>1000000</td>
</tr>
<tr>
<td>Owner: PSP1</td>
</tr>
</tbody>
</table>
And an ending note on token selection

PSP1 Admin

PSP1 UserA

3500000 not available!
Try Again!

Withdraw 50

<table>
<thead>
<tr>
<th>PSP1 Corda</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Token#1</strong></td>
</tr>
<tr>
<td>2000000</td>
</tr>
<tr>
<td>Owner: PSP1</td>
</tr>
<tr>
<td><strong>Token#2</strong></td>
</tr>
<tr>
<td>1000000</td>
</tr>
<tr>
<td>Owner: PSP1</td>
</tr>
<tr>
<td><strong>Token#3</strong></td>
</tr>
<tr>
<td>1000000</td>
</tr>
<tr>
<td>Owner: PSP1</td>
</tr>
</tbody>
</table>
And an ending note on token selection

PSP1 Admin

3500000 not available!
Try Again!

PSP1 UserA

Got 50

<table>
<thead>
<tr>
<th>Token#1</th>
<th>2000000</th>
<th>Owner: PSP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token#2</td>
<td>1000000</td>
<td>Owner: PSP1</td>
</tr>
<tr>
<td>Token#3</td>
<td>1000000</td>
<td>Owner: PSP1</td>
</tr>
<tr>
<td>Token#4</td>
<td>999950</td>
<td>Owner: PSP1</td>
</tr>
<tr>
<td>Token#5</td>
<td>50</td>
<td>Owner: UserA</td>
</tr>
</tbody>
</table>
Backchain and privacy

To be able to verify authenticity of the tokens all historic transactions for that token is needed.
Backchain and privacy

To be able to verify authenticity of the tokens all historic transactions for that token is needed.
Backchain and privacy

So PSP2 can see how PSP1 have done the issue and how the UserA withdrawn 200.
Backchain and privacy

Older and longer backchains reveals more.

PSP2 Admin
Backchain and privacy

Older and longer backchains reveals more.
Backchain and privacy

Older and longer backchains reveals more.
Backchain and privacy

Older and longer backchains reveals more.
Backchain and privacy

Older and longer backchains reveals more.
Backchain and privacy

Older and longer backchains reveals more.
Backchain and privacy

Older and longer backchains reveals more.
Backchain and privacy

Older and longer backchains reveals more.
Practical example: PSP2 backchain

Admin of PSP2 has only information from the PSP2 Corda node and business layer of PSP2.

To be able to get the backchain and to visualize it PSP2 admin has to:

• Extract the backchain

• Get the transactions

• Datamine

• Visualize
Practical example: Extract the backchain

Login to PSP2 Corda node and start the Corda node shell.

Run the command

```bash
run internalVerifiedTransactionsSnapshot
```
Practical example: PSP2 backchain

- wire:
  id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
  inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDB3699F50F4AC8FD2EB7291BD782DF5C53"
    - index: 0
  outputs:
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
      amount: "100.85 SEK issued by Riksbanken"
      holder:
        "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3JtbTGL7RG71TWPEaZJhNFKWZWRp7jCHtRqYdZshmAv1tawKDD55qDnXDFmkUSvMqQhaRdxAMPYinLSoP88JwAPreBZJw"
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
      amount: "5399.15 SEK issued by Riksbanken"
      holder:
        "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3JtbTFUvZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7ywYsDiBWLGYdmDh"
  commands:
    - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
      token:
        issuer: "O=Riksbanken, L=Stockholm, C=SE"
        tokenType:
        tokenIdentifier: "SEK"
      inputs:
        - 0
      outputs:
        - 0
        - 1
      signers:
        - "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3JtbTFUvZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7ywYsDiBWLGYdmDh"
Practical example: PSP2 backchain

- wire:
  - id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
  - txhash: "5C8618DCFB36BFABB0B6DB66331EBEBDB3699F50F4AC8FD2EB7291BD782DF5C53"
    - index: 0
  - inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEBDB3699F50F4AC8FD2EB7291BD782DF5C53"
  - outputs:
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
      - amount: "100.85 SEK issued by Riksbanken"
      - holder: "aSq9DsNNvGhYyYqqA9wd2eduEAZ5AXWg3TbTGL7RG71TWPcZJhnNFkWZWr7p7CHtRqYdZshmAvtawKDD55qDnXDFmkUSvMqQhaRdxMPYinLSo0p88JwAPReBZjw"
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
      - amount: "5399.15 SEK issued by Riksbanken"
      - holder: "aSq9DsNNvGhYyYqqA9wd2eduEAZ5AXWg3TbTFUZVr3NjFk7sDNTBjd3g3q9sJNbZKftVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnsBgkuVDw7yWYsDiBWLGYdMh"
    - commands:
      - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
        - token:
          - issuer: "O=Riksbanken, L=Stockholm, C=SE"
          - tokenType:
            - tokenIdentifier: "SEK"
      - inputs:
        - 0
      - outputs:
        - 0
        - 1
  - signers:
    - "aSq9DsNNvGhYyYqqA9wd2eduEAZ5AXWg3TbTFUZVr3NjFk7sDNTBjd3g3q9sJNbZKftVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnsBgkuVDw7yWYsDiBWLGYdMh"
Practical example: PSP2 backchain

- wire:
  - id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
  - txhash: "5C8618DCFB368FAB0B6DB66331EB6EB3699F50F4AC8FD2EB7291BD782DF5C53"
  - index: 0

- inputs:
  - txhash: "5C8618DCFB368FAB0B6DB66331EB6EB3699F50F4AC8FD2EB7291BD782DF5C53"

- outputs:
  - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
    amount: "100.85 SEK issued by Riksbanken"
    holder: "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3TbTGL7RG71TWEaZJhNFkWZWRp7jCHTqYdZshmA1tawKDd55qDnXDFmkUSvMqQhaRdxAMPyinLSop88JwAPReBZJw"

  - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
    amount: "5399.15 SEK issued by Riksbanken"
    holder: "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3TbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFbpgkuvDV7yWYsDiBWLGYdmDh"

- commands:
  - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
    token:
      issuer: "O=Riksbanken, L=Stockholm, C=SE"
      tokenType:
        tokenIdentifier: "SEK"
      inputs:
        - 0
      outputs:
        - 0
        - 1
    signers:
      "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3TbTFUZVr3NjFk7sDNTBjdg3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFbpgkuvDV7yWYsDiBWLGYdmDh"
Practical example: PSP2 backchain

- wire:
  id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864E4E28269E8F2F038"
  inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDEB3699F50F4AC8FD2EB7291BD782DF5C53"
      index: 0
  outputs:
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
      amount: "100.85 SEK issued by Riksbanken"
      holder: "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3TbTGL7RG71TWPEaZJhNFkWzWrp7jCHtrqYdZshmAvtawKd55qDnXDFmkUsvMqQhaRdxamPYinLSop88JwAPReBZJw"
    - data: !<com.r3.corda.lib.tokens.contracts.states.FungibleToken>
      amount: "5399.15 SEK issued by Riksbanken"
      holder: "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3TbTFUZVr3NjFk7sDNTBjd3g3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7ywZYsDiBWLGYdmDh"
  commands:
    - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
      token:
        issuer: "O=Riksbanken, L=Stockholm, C=SE"
        tokenType:
          tokenIdentifier: "SEK"
      inputs:
        - 0
      outputs:
        - 0
        - 1
      signers:
        - "aSq9DsNNvGhYxYyqA9wd2eduEAZ5AXWg3TbTFUZVr3NjFk7sDNTBjd3g3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7ywZYsDiBWLGYdmDh"
Practical example: PSP2 backchain

- wire:
  - id: "6BE4262593EA89C5097FED35221CC0A27FE78F7BB6C10864AE4E28269E8F2F038"
  - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDB3699F50F4AC8FD2EB7291BD782DF5C53"
  - index: 0
  - inputs:
    - txhash: "5C8618DCFB36BFABB0B6DB66331EBEDB3699F50F4AC8FD2EB7291BD782DF5C53"
  - outputs:
    - amount: "100.85 SEK issued by Riksbanken"
    - holder: "aSq9DsNNvGhYxYqqA9wd2eduEAZ5AXwg3TbTGL7RG71TWPEaZJhNFKWZWRp7jCHtRqYdZshmAvtawK5d5sDnXDFmkUSvMqQhaRdxaMPYinLSop88JwAPReBZJw"
    - amount: "5399.15 SEK issued by Riksbanken"
    - holder: "aSq9DsNNvGhYxYqqA9wd2eduEAZ5AXwg3TbTFUZVs3NjFk7sDNTBjd3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7ywYsDiBWLGYdmDh"
  - commands:
    - value: !<com.r3.corda.lib.tokens.contracts.commands.MoveTokenCommand>
      - token:
        - issuer: "O=Riksbanken, L=Stockholm, C=SE"
        - tokenType:
          - tokenIdentifier: "SEK"
  - signers:
    - "aSq9DsNNvGhYxYqqA9wd2eduEAZ5AXwg3TbTFUZVs3NjFk7sDNTBjd3q9sJNbZKfTVhDQ8vcyisu9mWsoMPA1Heqbb3ZbNirZFnBpgkuVDW7ywYsDiBWLGYdmDh"
Practical example: Get the transactions

Admin of PSP2 has now extracted and created a JSON list of all transactions for all backchains for all tokens on PSP2 Corda node.

```json
{  "edges": [  {  "source": {"id": "PSP:GfHq2tTVk9z4eXgyKAMEqYFMYACZy4RQAuN3p72MxBywj86qJnnk3EhzaNPr", "label": "PSP:GfHq2tTVk9z4eXgyKAMEqYFMYACZy4RQAuN3p72MxBywj86qJnnk3EhzaNPr"},  "target": {"id": "aSq9DsNNvGhYxYyq9wd2eduEAY5AXWg3TbKnaoNTewVAC9a27PCxXFdoS3pqhMa5duj6jJGEspqvvtx59oNehuLxgXVWwu330uURRezoeTogZjBqpAPFXkmKnC4", "label": "aSq9DsNNvGhYxYyq9wd2eduEAY5AXWg3TbKnaoNTewVAC9a27PCxXFdoS3pqhMa5duj6jJGEspqvvtx59oNehuLxgXVWwu330uURRezoeTogZjBqpAPFXkmKnC4"},  "value": "1337.30"  },  {  "source": {"id": "aSq9DsNNvGhYxYyq9wd2eduEAY5AXWg3TbKnaoNTewVAC9a27PCxXFdoS3pqhMa5duj6jJGEspqvvtx59oNehuLxgXVWwu330uURRezoeTogZjBqpAPFXkmKnC4", "label": "aSq9DsNNvGhYxYyq9wd2eduEAY5AXWg3TbKnaoNTewVAC9a27PCxXFdoS3pqhMa5duj6jJGEspqvvtx59oNehuLxgXVWwu330uURRezoeTogZjBqpAPFXkmKnC4"},  "target": {"id": "aSq9DsNNvGhYxYyq9wd2eduEAY5AXWg3TbTJX1eBDU2mve7qqDwomboVu9HDG1psQn5ss4TsE68tDuWKSRZ9hnJDrmcfxZ4tagpD7qM2UsvAGcPwbwG3qdAuts", "label": "aSq9DsNNvGhYxYyq9wd2eduEAY5AXWg3TbTJX1eBDU2mve7qqDwomboVu9HDG1psQn5ss4TsE68tDuWKSRZ9hnJDrmcfxZ4tagpD7qM2UsvAGcPwbwG3qdAuts"},  "value": "1337.40"  }]}
```
Practical example: Datamine

Admin of PSP2 has the public keys of users in the backchain but can enrich the information with wallet ID or alias with information in the logs or from user transaction log in the PSP2 business layer. To be able to do this any user on PSP2 must have done a transaction with that public key earlier.

User history record extract from PSP2 business layer.

<table>
<thead>
<tr>
<th>TxID</th>
<th>0DDB759E02091B3A52D61194AE7D464F7022DBF3DBAF45005B74F27F0E49A0B7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command</td>
<td>PAY</td>
</tr>
<tr>
<td>Payer wallet ID</td>
<td>9021f73d-d883-4cfb-8899-203690bff93f</td>
</tr>
<tr>
<td>Payer PSP</td>
<td>PSP1</td>
</tr>
<tr>
<td>Payee wallet ID</td>
<td>cf960d62-b1a6-4816-8179-717808d160d8</td>
</tr>
<tr>
<td>Payee PSP</td>
<td>PSP2</td>
</tr>
<tr>
<td>Amount</td>
<td>5.00</td>
</tr>
</tbody>
</table>
Practical example: Visualize

Just pour the enriched JSON transaction list into HTML with D3.js (https://d3js.org/) function d3.layout.force() to connect all the transactions.

```script
<d3.json("back2.json", function(error, data) {
  ... 
  var force = d3.layout.force();
  ... 
  graph = new myGraph();
  graph.initialize();

</script>
```
 Visualization with data only from PSP2
Backchain and privacy

Need to be compliant with:

• European General Data Protection Regulation (GDPR)

• Swedish bank secrecy regulation
Everything else

Everything that must be solved before going in production with a token based retail CBDC

• Performance and authenticity of the digital currency.

• High availability and in memory token selection.

• Catastrophic failures and disaster recovery.

• A secure offline?

• Non-repudiation.

• Information security (ISO 27000)
  • IT security (NIST, OWASP)
  • Laws, regulations and financial compliance
Solutions

There are many solutions for the presented challenges.

• Chain snipping, Chipping, Key rotation, Zero knowledge proof and other encryption.

• Validating notary node.

• Hardware wallets (e.g. smart cards).

• Restore procedures and functions for correcting inconsistencies.

The Riksbank is now experimenting with other designs and will also look at other technologies.
Summary

The goal of this presentation is to share insights of the security challenges of building a prototype of a two tier retail central bank digital currency based on a blockchain with value based tokens.

Only presented threats, vulnerabilities, security fails and some unknowns.

Not presented the good design and all the positive lessons learned!
Thank you for attending