UNIVERSITY OF TRENTO

Complexity, Cryptography, and Financial Technologies

Lecture 2 – Introduction to Crypto-based FinTech

Chan Nam Ngo

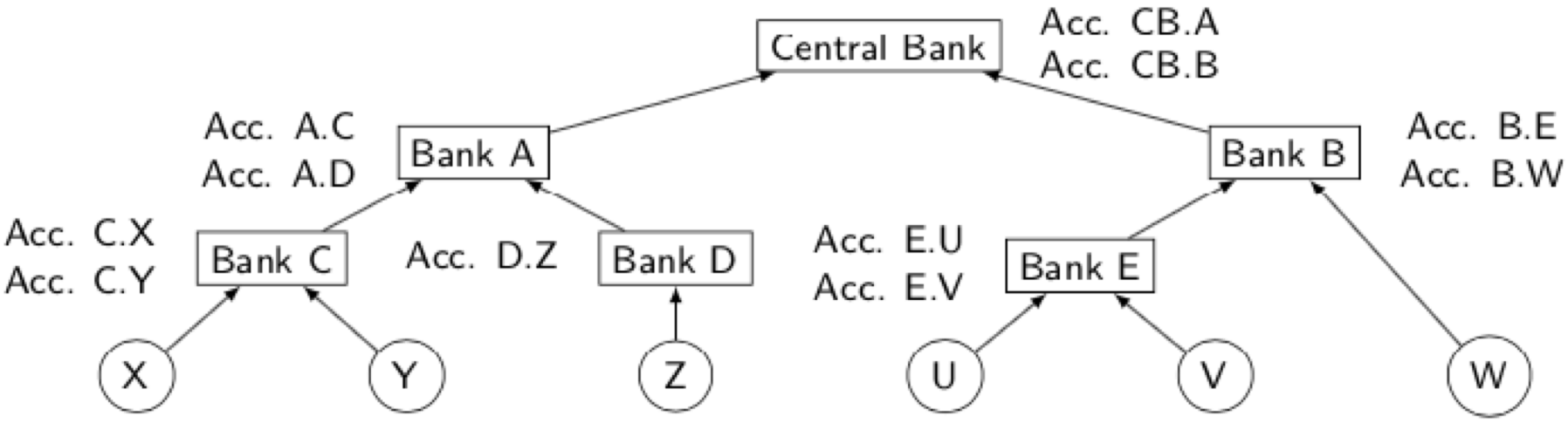
Let's focus on Payment Transaction Network (PTN)

UNIVERSITY OF TRENTO

- Traditional PTN
- Crypto-based PTN
- The high-level features of PTN
- Security of crypto-based PTN
  - Security Requirements
  - Threats and Countermeasures
- DigiCash (1990) as an example

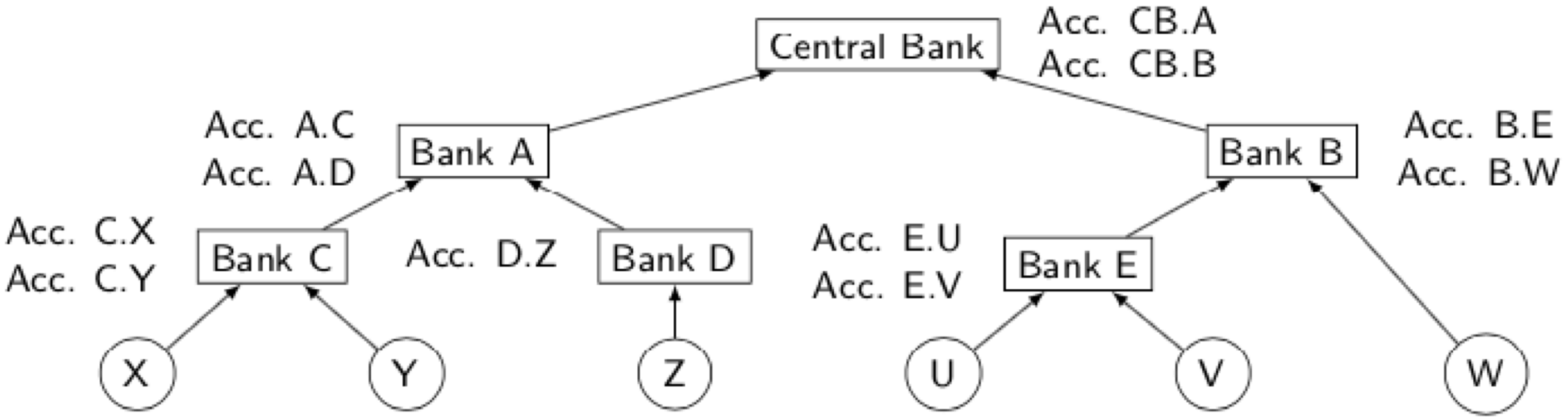
Traditional PTN

- **ALL clients transact through SOME central authorities (CA)**
- **Mostly hierarchal structure**



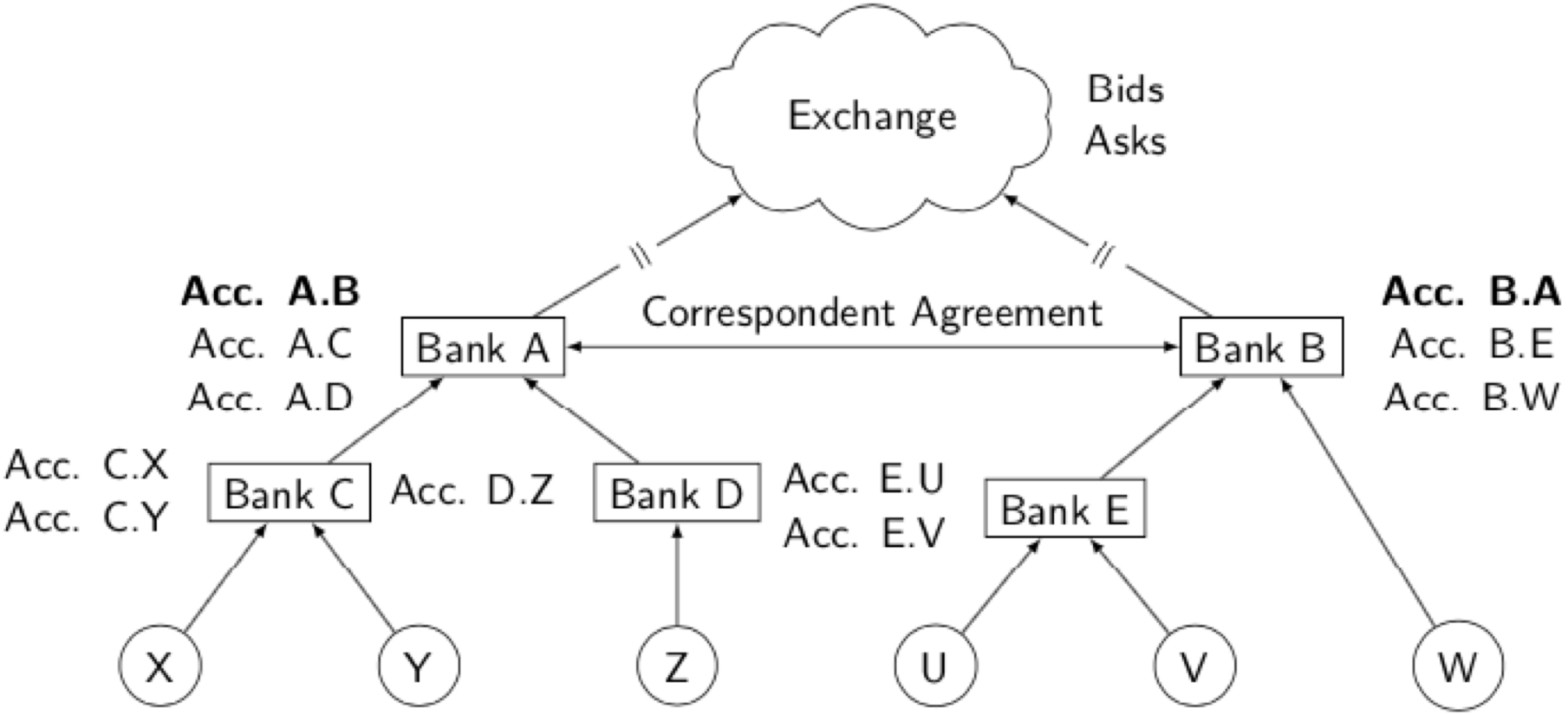
Traditional PTN – Transaction Path

- Client X is with Bank C
- Client V is with Bank E
- A transaction X → V
- actually means transactions C → A → CB → B → E



Traditional PTN – International Transaction

- **There could be an Exchange if it involves international transactions**



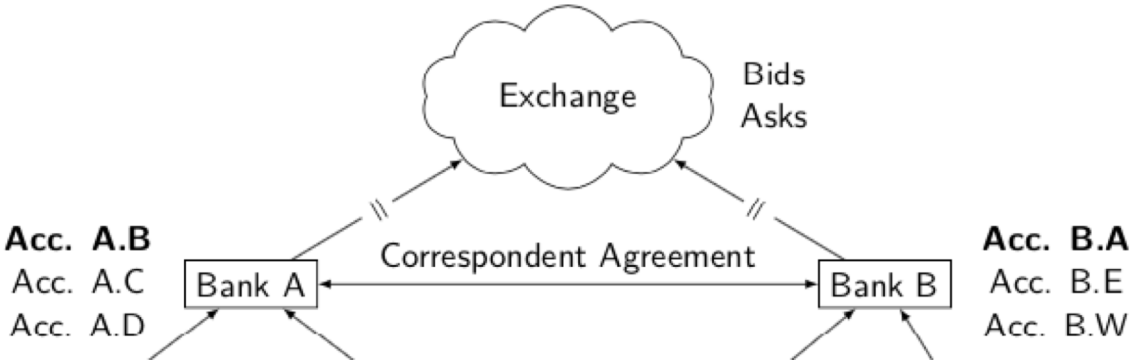
Traditional PTN – Nostro/Vostro Accounts

UNIVERSITY OF TRENTO

- **Correspondence account**
  - Held by a bank to make transactions on behalf of another bank, usually oversea
- **Nostro account**
  - Our money

  held by them (another bank)
- **Vostro account**
  - Their money

  held by us



Traditional PTN Examples

UNIVERSITY OF TRENTO

- **Daily payment**
  - Direct Payment: cash
  - Indirect Payment: ATM, Debit, Credit cards
  - ”Wrappers”: PayPal, Google Wallet, Apple Pay
- **Gross Payment**
  - Target2 (EU)
  - FedWire (US)
  - BACS (UK)

Crypto-based PTN

UNIVERSITY OF TRENTO

- **ALL clients transact/trade DIRECTLY**
- **Or through SOME "semi/un-trusted" servers**
  → Require a Security Protocol
- **Privacy/Anonymity**
  - Not ALWAYS
    - Bitcoin provides no real anonymity, transactions graph can be rebuilt from pseudonyms
    - http://learningspot.altervista.org/how-to-de-anonymize-bitcoin/
  - But POSSIBLE:
    - ZCash provides REAL privacy for transactions

Crypto—based PTN history (1)

UNIVERSITY OF TRENTO

- **DigiCash (1990):**
  - Exchange fiat money into digital “coins” to spend
  - Can only transact if Payer and Payee both have accounts at the same centralized DigiCash Bank
  - Use Blind Signature for coins authenticity and client anonymity
  - https://en.wikipedia.org/wiki/DigiCash
- **Some (unsuccessful) attempts**
  - Bit-Money, Reusable Proof Of Work (RPOW), Bit-Gold
- **Bitcoin (2009)**
  - Decentralized PTN
    - with a public ledger maintained by many nodes
    - These nodes are called the Bitcoin miners
  - Proof-of-Work (Computational-costly Hash Function)
    - for consensus and value creation
    - Finding a PoW is called mining for Bitcoin
  - Blockchain (the data structure of the public ledger)
  - Use Digital Signature for coins authentication

Crypto—based PTN history (2)

- **Bitcoin variants**
  - DogeCoin, LiteCoin, PotCoin
    - Use memory-costly Hash Functions
    - to deter Application-Specific Integrated Circuit (ASIC) mining
  - BlackCoin, Nxt
    - Proof-of-Stake: miners ”bet” on the validity of the transactions
- **Some real advances in the field**
  - Ethereum
    - programmable cryptocurrency,
    - allows the building of complex FinTech,
    - no privacy
  - ZCash
    - privacy-preserving cryptocurrency

Four PTN High-level Actors

UNIVERSITY OF TRENTO

1. **Payer – who PAYs**
2. **Payee – whom to be PAID**
3. **Brokers – untrusted intermediaries**
4. **Central Authority (CA) – recognized and trusted intermediary**
   - Centralized PTN: only CA decides on transactions validity
     - DigiCash
   - Decentralized PTN: Brokers collectively decide on transactions validity
     - Bitcoin, Zcash, Ethereum
   - Hybrid PTN: CA and Brokers share decisions
     - Ripple, RSCoin

A standard payment procedure summarized

UNIVERSITY OF TRENTO

- **After a sender submits a payment message to a payment system, the message must pass through that system's validation procedures.**
- **Validation will vary by system and can include security measures, such as**
  - verification of the sender's identity
  - and the integrity of the message.
  - […] the availability of sufficient funds or credit for settlement.
- **Payments that pass the conditionality test are prepared for settlement.**
- **Under some payment system frameworks, settlement finality (that is, when settlement is unconditional and irrevocable) occurs when the receiver's account is credited.**

Source: Millers et. al., Distributed ledger technology in payments, clearing, and settlement, 2016

A standard payment procedure summarized (2)

- **Two required steps can be extracted:**
  - After a sender submits a payment message to a payment system […]
    → A promise to pay
  - […] settlement finality (that is, when settlement is unconditional and irrevocable), […]
    → Promise is fulfilled
- **What is missing?**
  - Where does the transacted value come from? Who put new value into circulation?
  - Where is the value (or the payment history) stored?

Four PTN High-level Conceptual Steps

UNIVERSITY OF TRENTO

1. **Creation of Value**
   - New value is added into the network for circulation
2. **Promise of Payment**
   - Payer announces that she wants to pay a Payee X amount
3. **Fulfillment of Transactions**
   - The Payer is debited X amount and the Payee is credited X amount
4. **Preservation of Value**
   - The debits and credits go into the public ledger

Token-based vs Account-based PTN

- **Token-based**
  - A token, normally called “coin”, represents some tradable value
  - A user keeps a “wallet” which stores “coins”
  - A transaction from a Payer to a Payee is a transfer of “coins” between them
- **Account-based**
  - value is stored as a pair of (user, balance) in a “Bank”
  - A transaction is a debit of the Payer’s account and credit of the Payee’s account
- **Transaction log is normally kept for audit**

1. Creation of Value

- **Payer**
  - out-of-band deposits value into CA/Brokers
  - by exchanging real world fiat money into PTN value
  - or value is rewarded to Payer after doing some "work" such as "solving a challenge"
- **CA/Brokers**
  - Credit Payer's account balance (in Account-based)
  - Or send new "coins" to Payer (in Token-based)

2. Promise of Payment

- **Payee**
  - Sends Payee ID (and the Amount) to Payer
- **Payer**
  - Receives Payee ID (and the Amount)
  - Creates a transaction (Payer ID, Payee ID, Amount)
    - Or (Payer ID, Payee ID, Amount, Coins)
    - Or (Payee ID, Amount, Coins) ← Why is this possible???
    - Or (Payee ID, Coins) ← Can we do only this???

    → Transaction data can be varied by systems
  - "Signs" the transaction
  - Sends the "signed" transaction to CA/Brokers
    - Or the signed transaction can go through Payee to CA/Brokers

    → Are the two cases different???
- **CA/Brokers**
  - Receives the transaction from the Payer/Payee

3. Fulfillment of Transactions

UNIVERSITY OF TRENTO

- **CA/Brokers**
  - Validate the transaction
    - Payer "signature" is valid
    - Payer has more than X amount in account
    - Or the coins in the transaction are authentic, unspent and greater than X in total
  - Fulfill the transaction
    - Payer is debited X amount and Payee is credited X amount
    - Or mark the old coins "spent" and send new coins to Payee
- **Payee**
  - Receives the new coins from CA/Brokers in Token-based PTN

4. Preservation of Value

- **CA/Brokers**
  - Store the accounts balance
  - Store the “spent” coins
  - Store the transaction history
- **Payer/Payee**
  - Store the “unspent” coins
  - Store the authentication secret

Security of Crypto-based PTN

UNIVERSITY OF TRENTO

- **Security requirements of a PTN**
  - Integrity: loss of value, fraud, theft
  - Confidentiality vs Anonymity
    - Confidential = know the owner but cannot see the value
    - Anonymous = can see the value but cannot know the owner
- **A security protocol that realizes a PTN must satisfies all the security requirements**

Threats to PTN Integrity

UNIVERSITY OF TRENTO

- **Loss of value**
  - The value is lost, cannot be circulated anymore
  - If it involves the victim, it is individual loss
  - Otherwise it is systemic loss
- **Fraud**
  - The victim is involved in a transaction that benefits another party with her own value
- **Theft**
  - The victim does not know about a transaction that involves her own value and benefits another party

Threats	Does another party benefit from this?	Is the victim actively involved?
Systemic Loss	-	-
Individual Loss	-	X
Fraud	X	X
Theft	X	-

Threats to PTN Integrity - Examples

UNIVERSITY OF TRENTO

- **Loss of value**
  - Individual Loss
    - Payer forgets the authentication secret or the signing key that is required to spend the value
  - Systemic Loss
    - CA is faulty or Brokers cannot reach consensus
- **Fraud**
  - Over-Drafting
    - Payer wants to pay more than her available fund
  - Double-Spending
    - Payer spends a coin twice
- **Theft**
  - Unauthorized-Spending
    - Payer wants to spend value/coin of ANOTHER Payer

Countermeasures for PTN Integrity

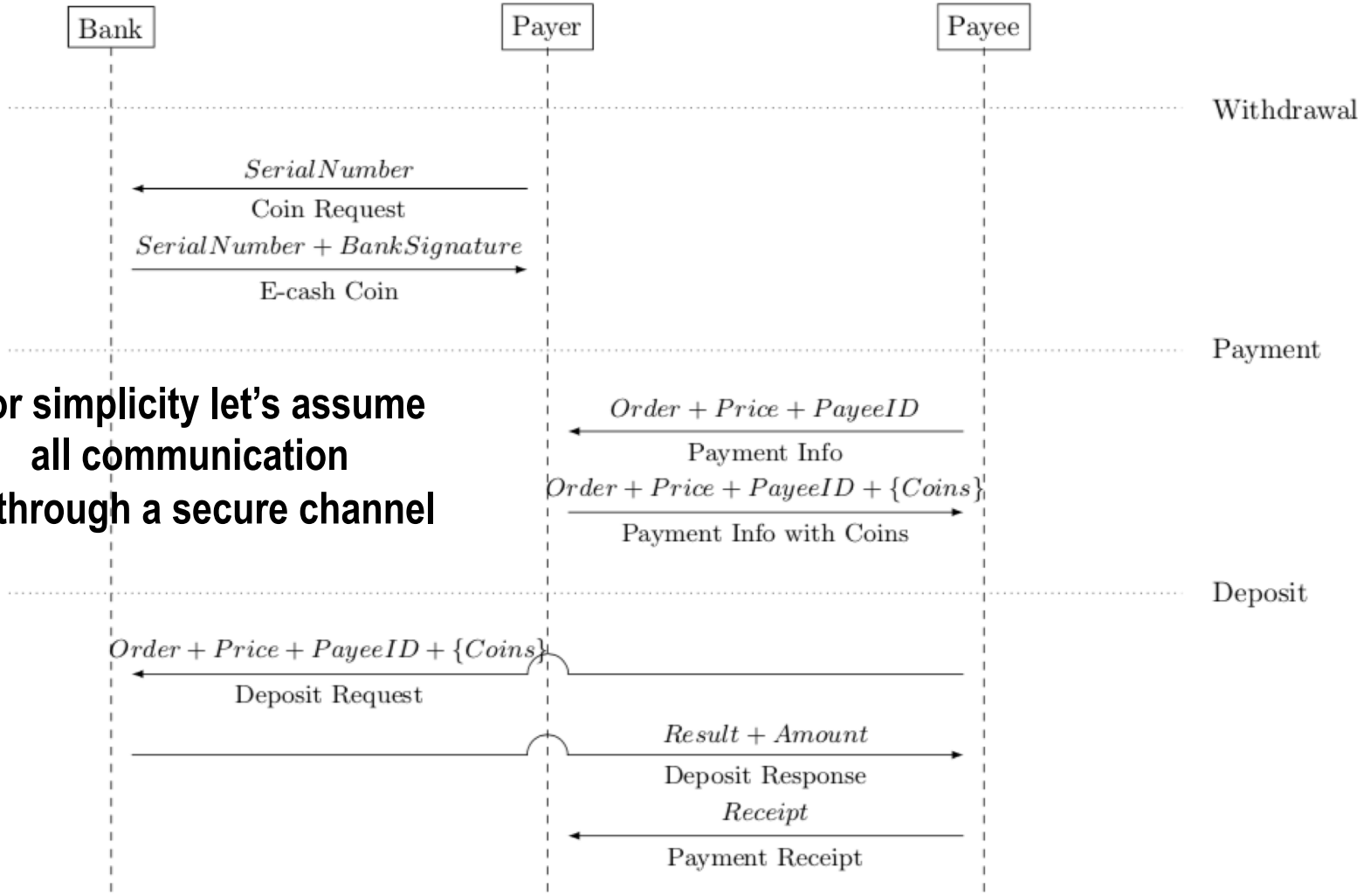
UNIVERSITY OF TRENTO

- **Over-Drafting**
  - CA/Brokers must check Payer's available fund when validating a transaction
- **Double-Spending**
  - CA/Brokers must mark the old coins as "spent" upon fulfilling a transaction
- **Unauthorized-Spending**
  - CA/Brokers must ask for a valid signature or an authentication secret of the spending "coins" when validating a transaction
- **Individual Loss**
  - Some backup mechanisms …
- **Systemic Loss**
  - N/A

Confidentiality vs Anonymity

- **Involves answering three questions**
  - Instantaneous Networth
    - At time t, can we identify the total value v of a nominal identity I?
  - Transient Value
    - At time t, can we know about a transaction of value v between two nominal identities I_1 and I_2?
  - Persistent Identity
    - Can we link two nominal identities I_1 at time t and I_2 at time t'?
- **Countermeasures are varied by PTN**

DigiCash (also called online E-cash)



For simplicity let's assume all communication is through a secure channel

How do we proceed?

UNIVERSITY OF TRENTO

- **We go a bit more technical, just a bit more**
- **If you don't understand at some point, e.g.**
  - I don't know what is a Digital Signature
  - I don't know the difference between Private Key Encryption and Public Key Encryption
- **stop me and ask**
- **I will provide some brief description**
- **so you can understand them as black boxes**

Public Key Encryption for Secrecy

- **Key Generation**
  - (pub, pri) ← KeyGen()
    - pub is called the public key
    - pri is called the private key
- **Encryption**
  - c ← Enc(pub,m)
    - m is the plaintext
    - c is called the ciphertext
- **Decryption**
  - m = Dec(pri,c)

Blind Signature for Anonymity

UNIVERSITY OF TRENTO

- **Blind Signature**
  - A client can obtain a signature from a server for a message m without the server knowing m.
- **5 algorithms**
  - Key Generation
    - (vk, sk) ← BKeyGen()
      - vk is called the verifying key
      - sk is called the signing key
  - Message Blinding
    - (x,r) ← Blind(vk,m)
      - m is the message to be signed
      - r is called the blinding factor
      - x is called the blinded message
  - Blind Signing
    - y ← Sign(sk,x)
      - y is called the blind signature
  - Signature Unblinding
    - s = Unblind(y,r)
      - s is the signature on m
  - Signature Verification
    - {0,1} ← Verify(vk,m,s)
      - Return 1 if s is a valid signature on m
      - Return 0 otherwise

Blind Signature for Anonymity - Setup

- **Bank**
  - (vk,sk) ← BKeyGen()
  - (pub,pri) ← KeyGen()
  - Broadcasts the verifying key vk and the public key pub
  - Stores the bank accounts of the Payer/Payee
- **Payer/Payee**
  - Receives/stores the verifying key vk and the public key pub
  - Opens an account with the Bank
    - Payer also needs to put some money into her account

Blind Signature for Anonymity – Coin Withdrawal

UNIVERSITY OF TRENTO

- **For simplicity let's assume a "coin" worths v$**
- **Payer**
  - Picks a random string m
  - (x,r) ← Blind(vk,m)
  - Sends x to the Bank
  - Receives y from the Bank
  - s = Unblind(y,r)
  - (m,s) is a "coin" to be stored by Payer
- **Bank**
  - Receives x from Payer
  - Checks if Payer has at least v$ in account
  - If YES
    - y ← Sign(sk,x)
    - Subtracts v$ from Payer's account
    - Returns y back to the Payer
  - Otherwise rejects the withdrawal request

Blind Signature for Anonymity – Payment and Deposit

UNIVERSITY OF TRENTO

- **Payer**
  - Gets Payee ID and amount p from Payee
    - Supposed p = v$
  - Creates a transaction t = (Payee ID, m, s, p)
  - Encrypts the transaction c ← Enc(pub, t)
  - Sends the encrypted transaction c to Payee
- **Payee**
  - Receives c from the Payer
  - Sends c to the Bank
- **Bank**
  - Receives c from Payee
  - Decrypts t = Dec(pri,c)
  - b = Verify(vk,m,s)
  - If b = 1, cont.
  - Checks if p = v
  - if YES, cont.
  - Checks if m is “spent”
  - If NO, cont.
    - Marks m as “spent”
    - Adds v$ into Payee’s account
  - Any check fails, rejects the deposit and notifies Payee

Exercise time!!!

- Identify the steps that are relevant to the 4 high-level conceptual steps
- Identify the steps that mitigate the threats
- Let's go back and see the protocol again

Suggested Readings

UNIVERSITY OF TRENTO

- **A book on E-Payment**
  - O'mahony, Donal, Peirce, Michael, and Tewari, Hitesh. *Electronic payment systems*. Boston, MA: Artech House, 1997.
- **DigiCash**
  - Chaum, David. *Blind signatures for untraceable payments*. In *Advances in cryptology*, pp. 199-203. Springer, Boston, MA, 1983.