Digital identity

How Bitcoin works?

Content

- How to establish a digital identity?

Problem 2 of e-Kuna

Who adds the transactions?

Alice

Ledger

Alice pays Bob $50

Alice pays Charlie $20

Bob pays Charlie $100

Bob

Charlie

Problem 2 of e-Kuna

Who adds the transactions?

Alice

Ledger

Alice pays Bob $50

Alice pays Charlie $20

Bob pays Charlie $100

Was it Alice that sent me the funds?

Bob

Charlie

Problem 2 of e-Kuna

Who adds the transactions?

Alice

Ledger

Alice pays Bob $50

Alice pays Charlie $20

Bob pays Charlie $100

Alice pays Bob $10000

Bob

Charlie

Digital identity

Alice

Bob

Digital identity

Alice

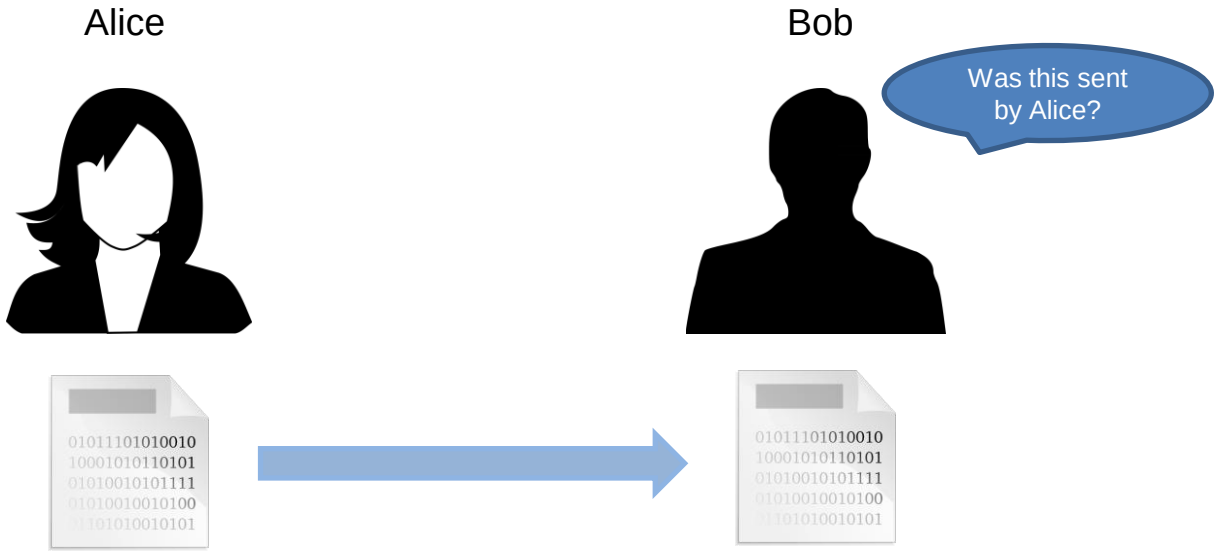
Bob

Digital identity

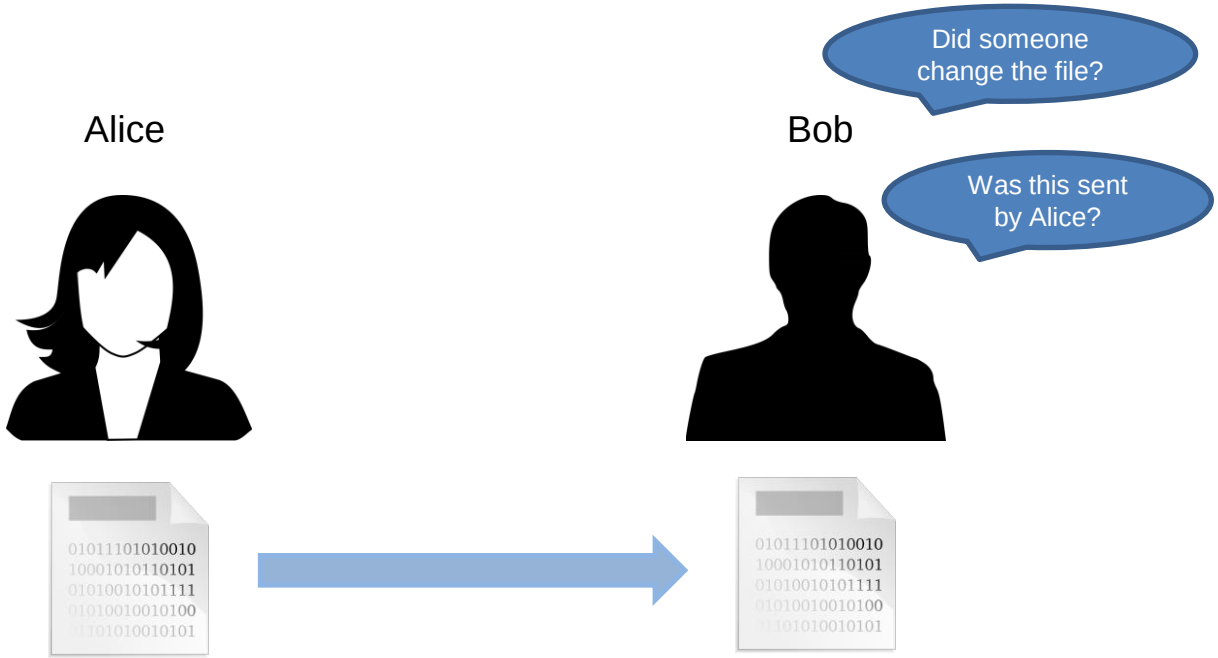
Alice

Bob

Digital identity



Digital identity



Solution in the analog world

Alice

Bob

Solution in the analog world

Alice

Bob

Alice

Solution in the analog world

Alice

Alice

Bob

Solution in the analog world

Alice

Bob

Alice

Does this work digitally?

Alice

Bob

Alice

Does this work digitally?

Alice

Bob

Alice = 110101

Does this work digitally?

Alice

Alice = 110101

Bob

Does this work digitally?

Alice

Bob

Alice = 110101

Does this work digitally?

Alice *Alice* = 110101

Bob

Charlie

Does this work digitally?

Alice

Alice = 110101

Bob

Charlie

Does this work digitally?

Alice

Alice = 110101

Bob

Charlie

Does this work digitally?

Alice

Alice = 110101

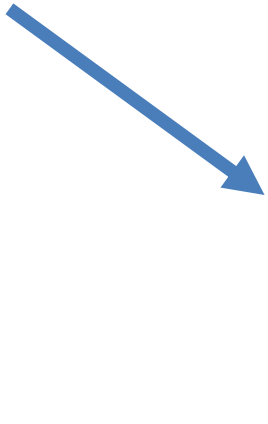
Bob

Charlie

Does this work digitally?

Alice

Alice = 110101



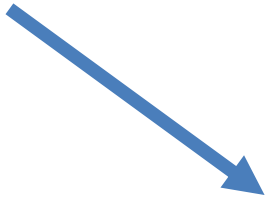
Bob

Charlie

Does this work digitally?

Alice

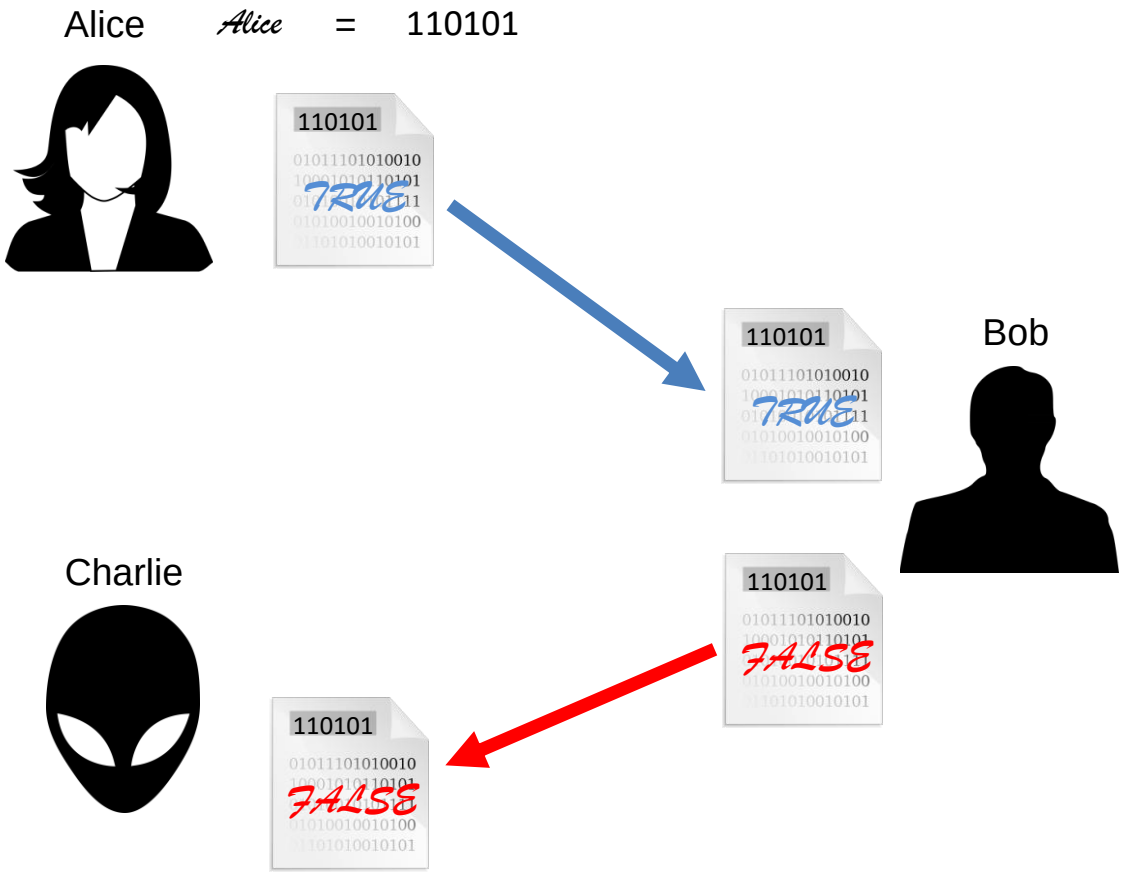
Alice = 110101



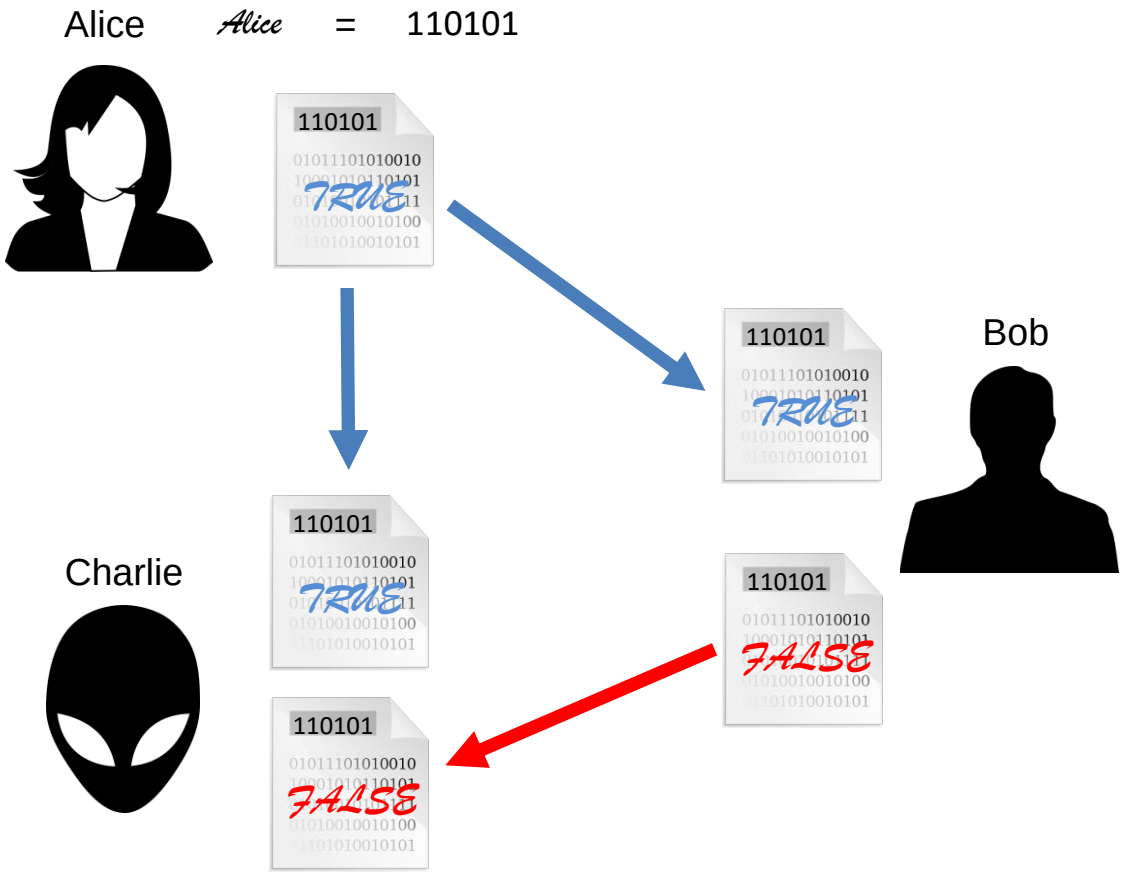
Bob

Charlie

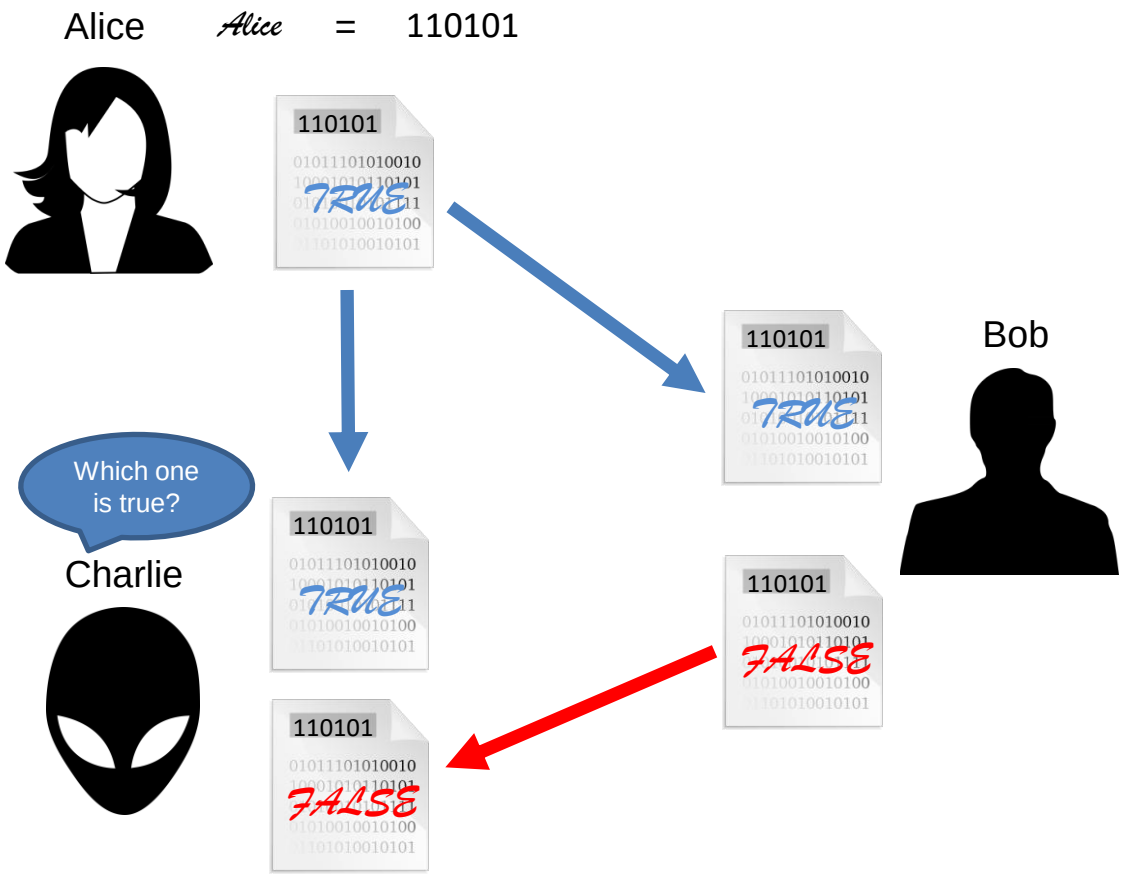
Does this work digitally?



Does this work digitally?



Does this work digitally?



Digital signatures

Our second cryptographic primitive

A digital signature protocol

- Only Alice can sign her own documents
- Each document has a different signature
- Anyone can check that Alice signed the document

Digital signatures

Our second cryptographic primitive

A digital signature protocol

- Only Alice can sign her own documents
- Each document has a different signature
- Anyone can check that Alice signed the document

Consists of three algorithms

1. Algorithm for generating public and secret keys
2. Algorithm for signing a document
3. Algorithm for verifying a signature

Digital signatures

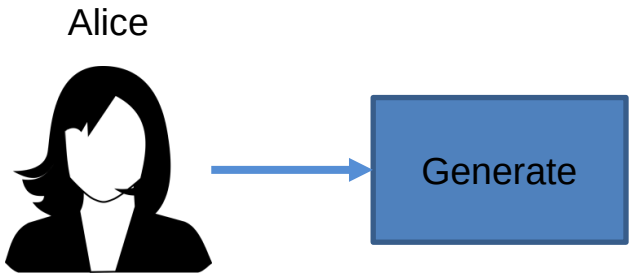
Generating keys

Alice

Generate

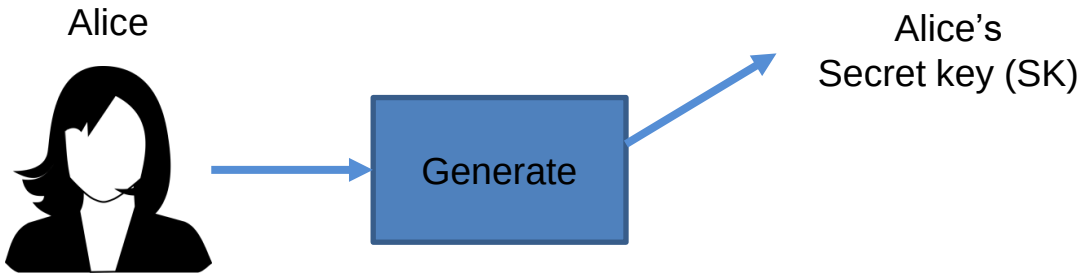
Digital signatures

Generating keys



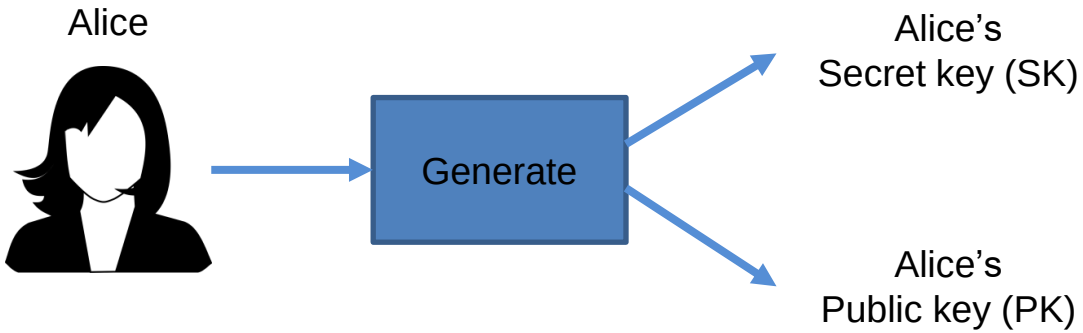
Digital signatures

Generating keys



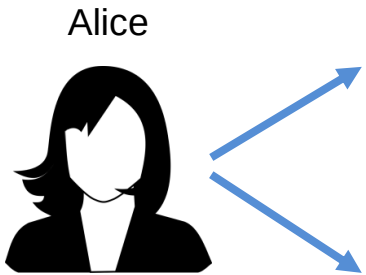
Digital signatures

Generating keys



Digital signatures

The Secret key and the public key



Alice's
Secret key (SK) = Only Alice knows this!

Alice's
Public key (PK) = Public information

Alice's digital identity

Digital signatures

Signing a document

Alice (SK,PK)

Digital signatures

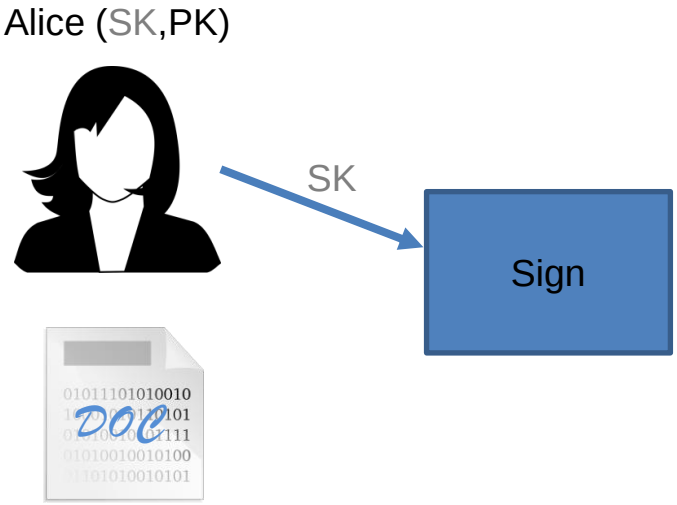
Signing a document

Alice (SK,PK)

Sign

Digital signatures

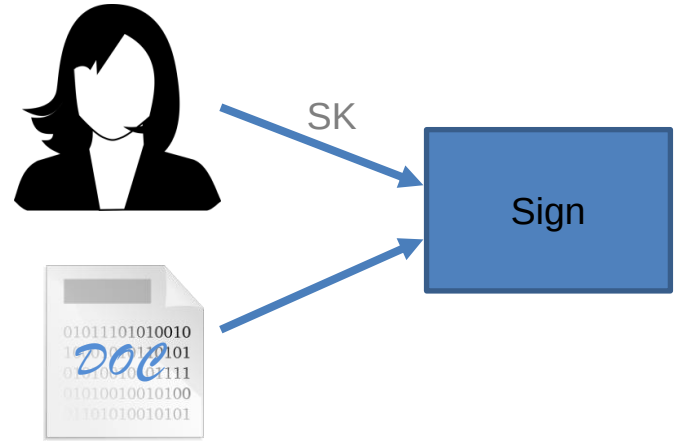
Signing a document



Digital signatures

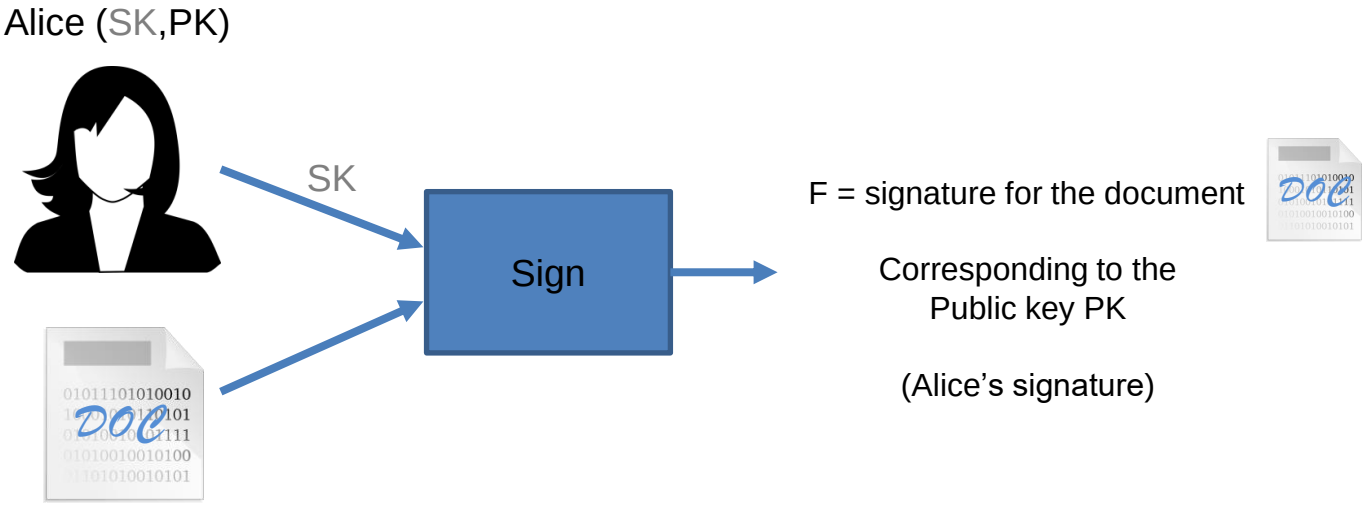
Signing a document

Alice (SK,PK)



Digital signatures

Signing a document



Digital signatures

Verifying digital signatures

Alice (SK,PK)

Bob

PK

Digital signatures

Verifying digital signatures

Alice (SK,PK)

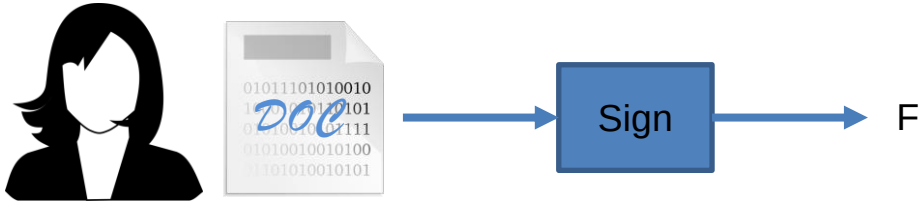
Bob

PK

Digital signatures

Verifying digital signatures

Alice (SK,PK)

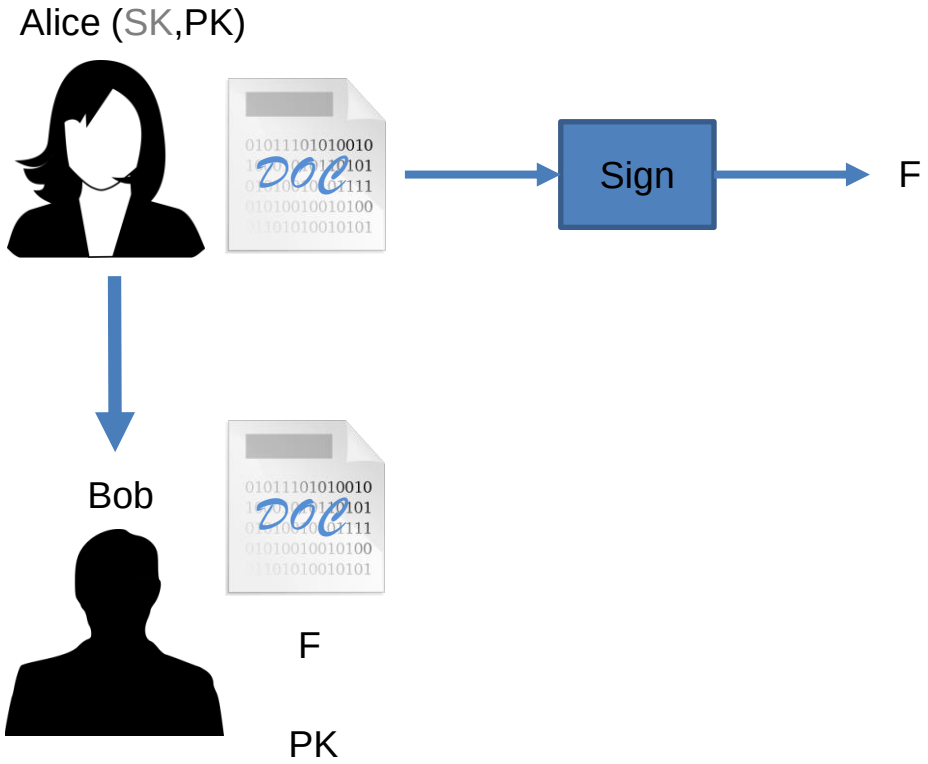


Bob

PK

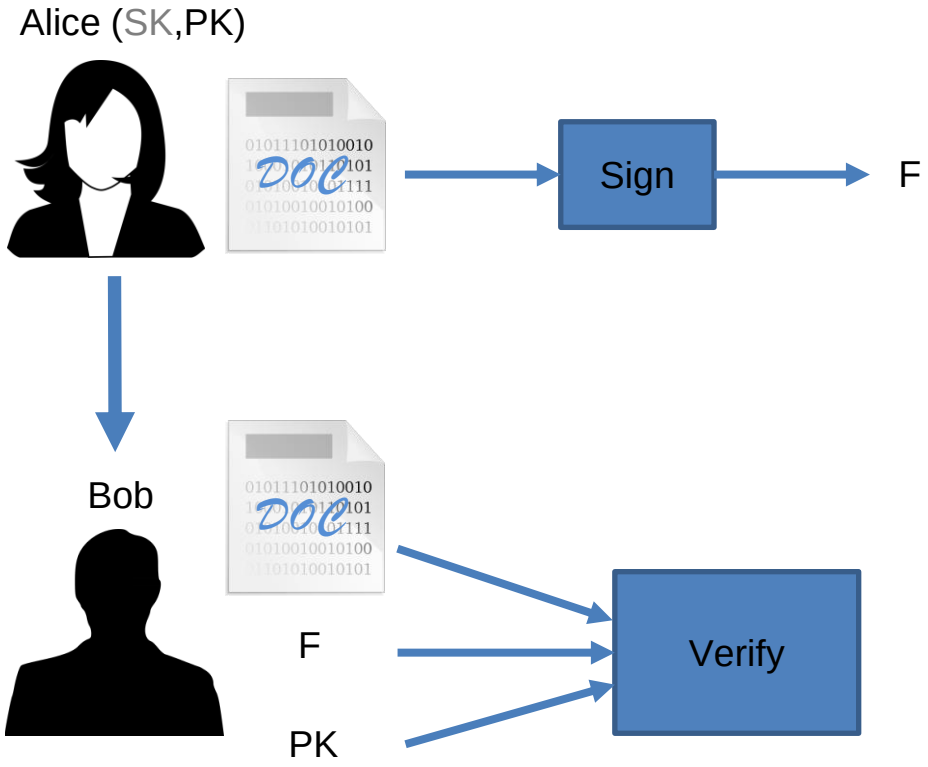
Digital signatures

Verifying digital signatures



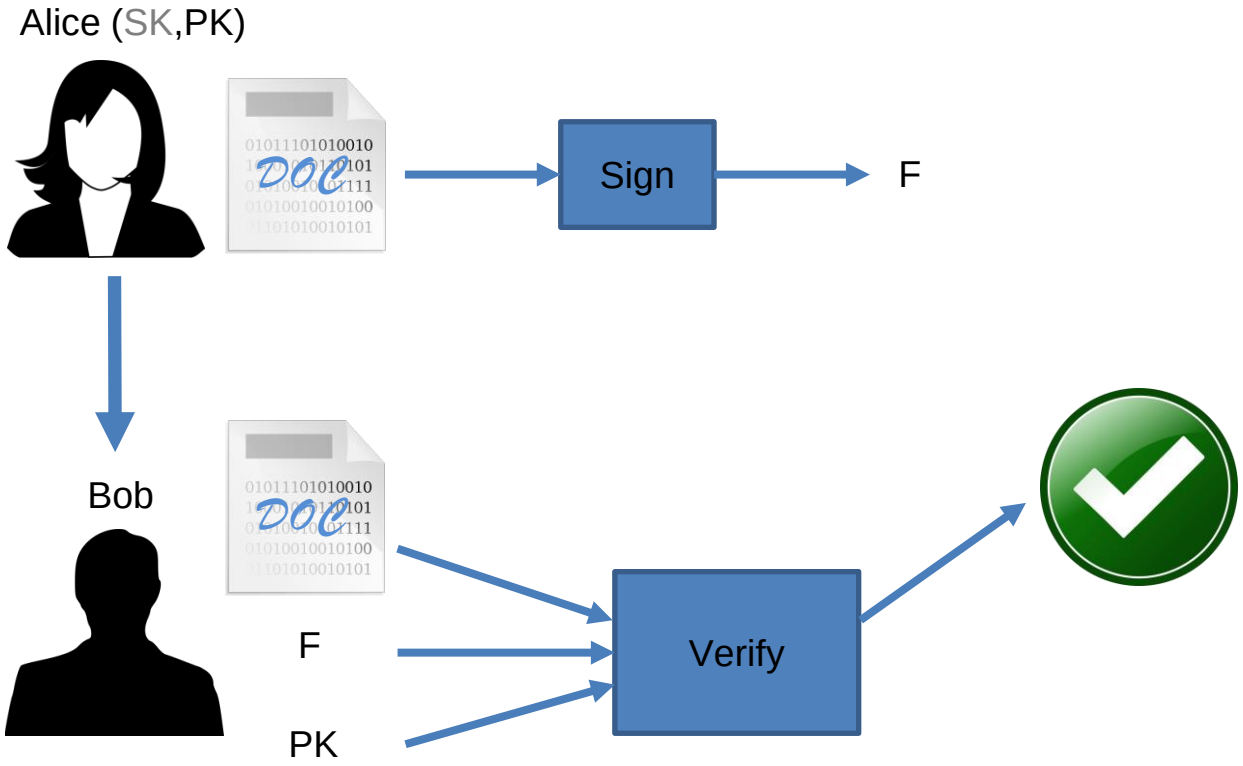
Digital signatures

Verifying digital signatures



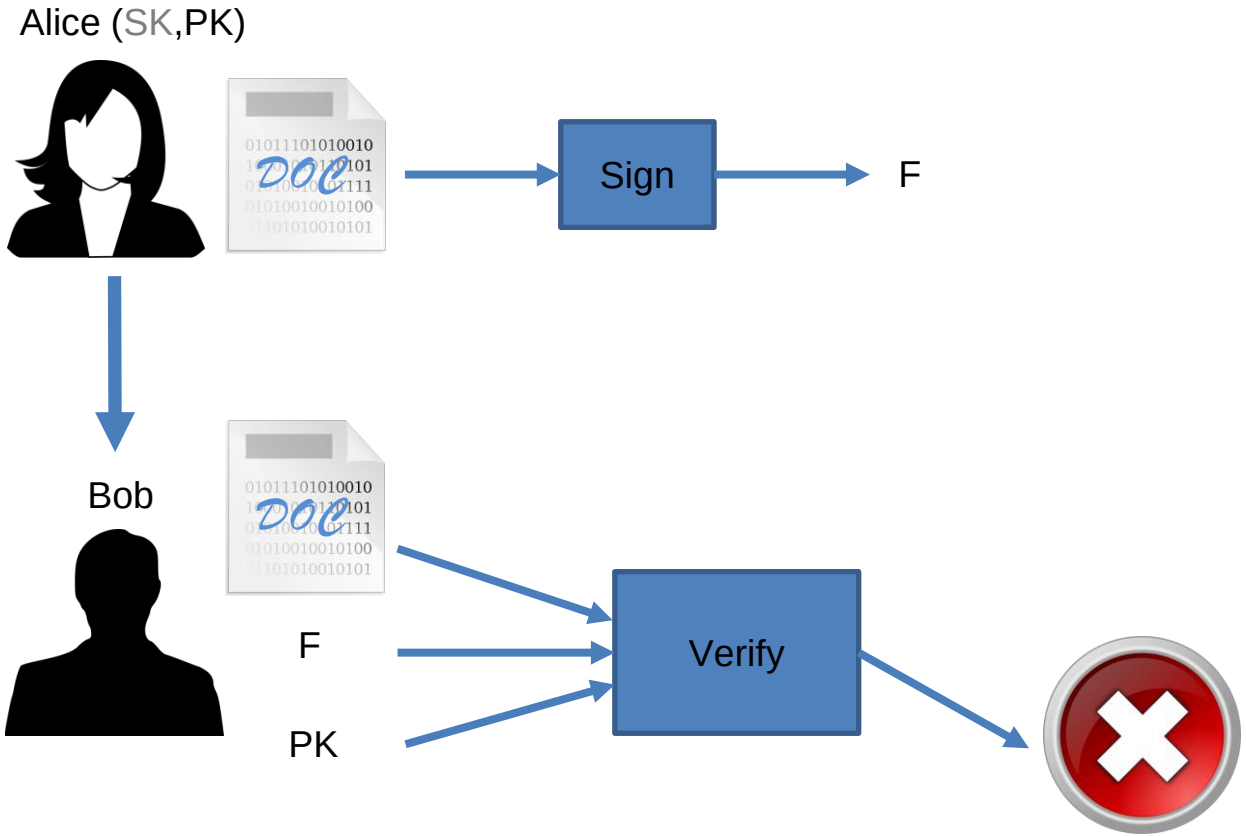
Digital signatures

Verifying digital signatures



Digital signatures

Verifying digital signatures



Properties of Digital signatures

- The three algorithms are safe:
  Impossible to reconstruct SK if we know PK
  Impossible to reconstruct SK if we know signed messages

- Signature is unfalsifiable and unique:
  Signature can only be realized using SK
  If the document changes, the signature changes

Properties of Digital signatures

Impossible to reconstruct SK if we know PK

Alice

(SK,PK)

Bob

PK

Properties of Digital signatures

Impossible to reconstruct SK if we know PK

Alice

(SK,PK)

Bob

PK

SK

Properties of Digital signatures

Impossible to reconstruct SK if we know signed messages

Alice

(SK,PK)

Bob

PK

Properties of Digital signatures

Impossible to reconstruct SK if we know signed messages

F1

Alice

F2

Bob

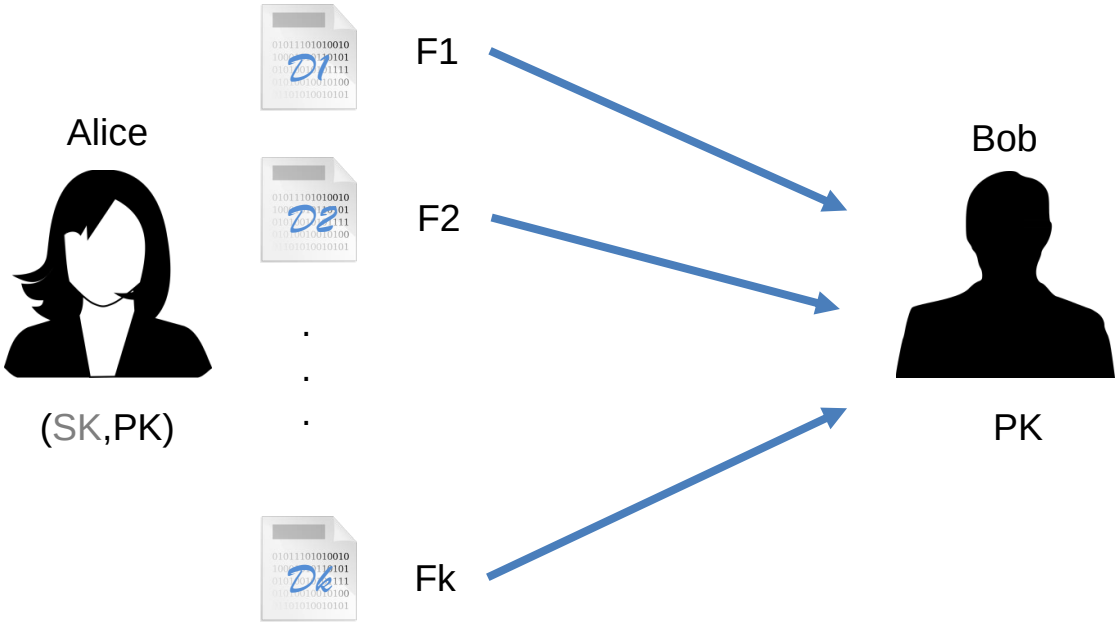
(SK,PK)

PK

Fk

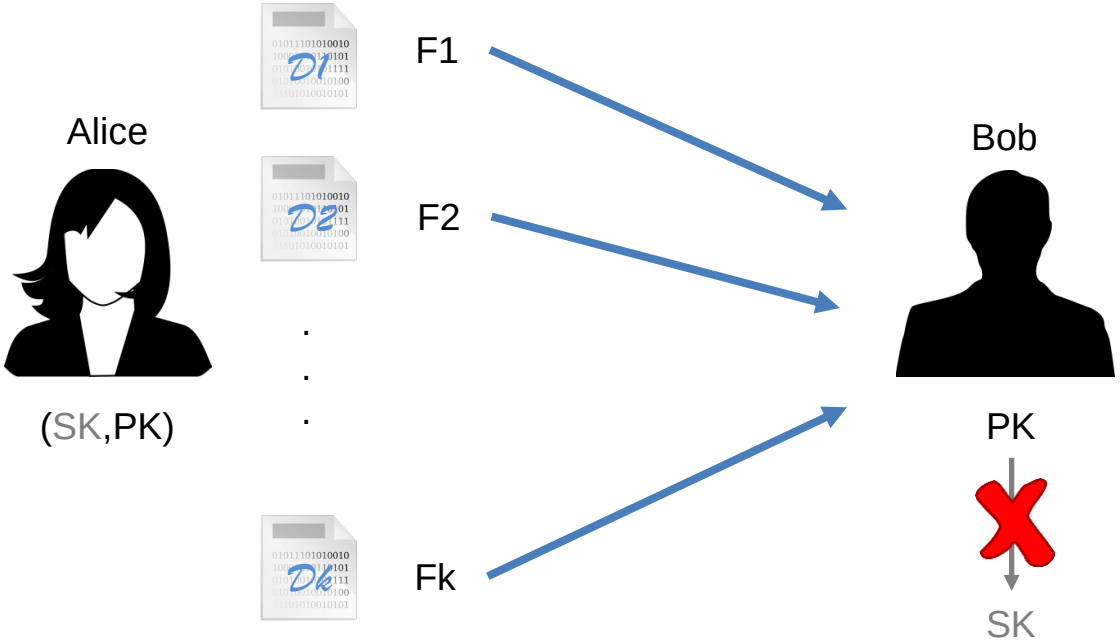
Properties of Digital signatures

Impossible to reconstruct SK if we know signed messages



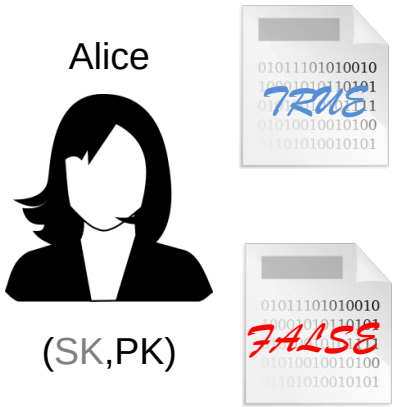
Properties of Digital signatures

Impossible to reconstruct SK if we know signed messages



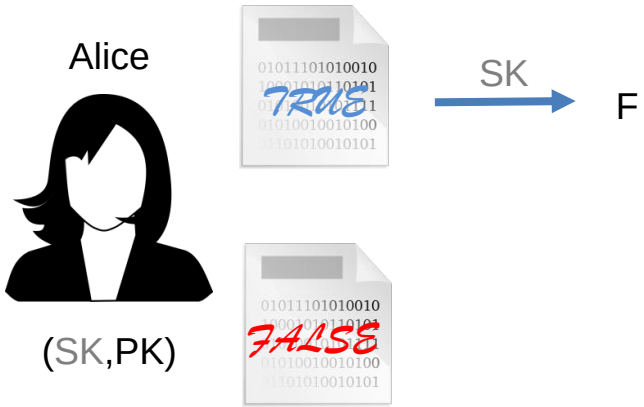
Properties of Digital signatures

The signature is unique for each document



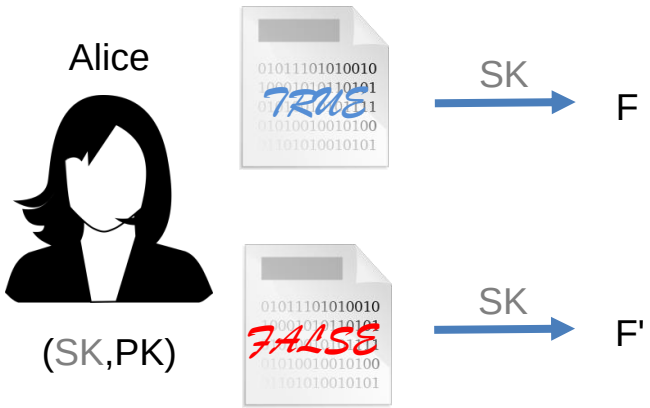
Properties of Digital signatures

The signature is unique for each document



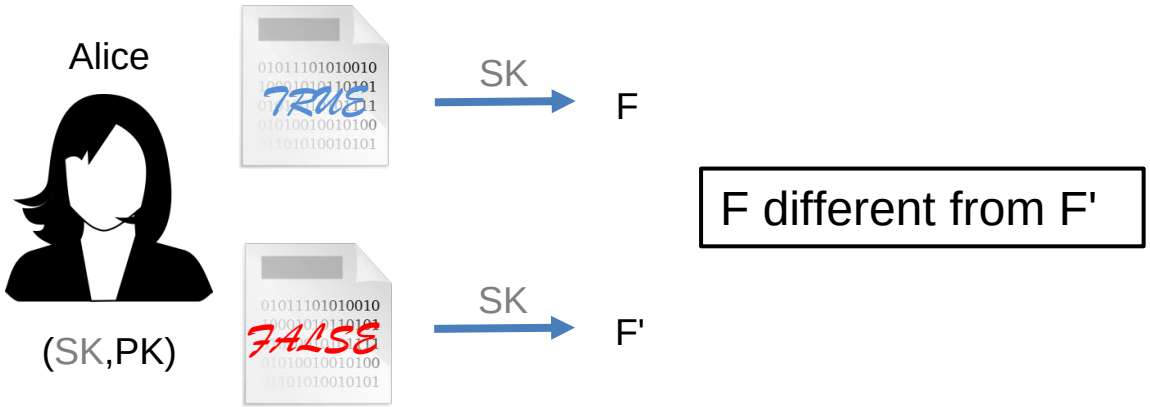
Properties of Digital signatures

The signature is unique for each document



Properties of Digital signatures

The signature is unique for each document



Properties of Digital signatures

We can sign only with SK

Alice
(SK,PK)

Charlie

Bob (SK',PK')

Properties of Digital signatures

We can sign only with SK

Alice
(SK,PK)

SK → F

Charlie

Bob (SK',PK')

Properties of Digital signatures

We can sign only with SK

Alice
(SK,PK)

SK → F

Charlie

Bob (SK',PK')

Properties of Digital signatures

We can sign only with SK

Alice
(SK,PK)

SK → F

Charlie

Bob (SK',PK')

SK' → F'

Properties of Digital signatures

We can sign only with SK

Alice
(SK,PK)

SK → F

Charlie

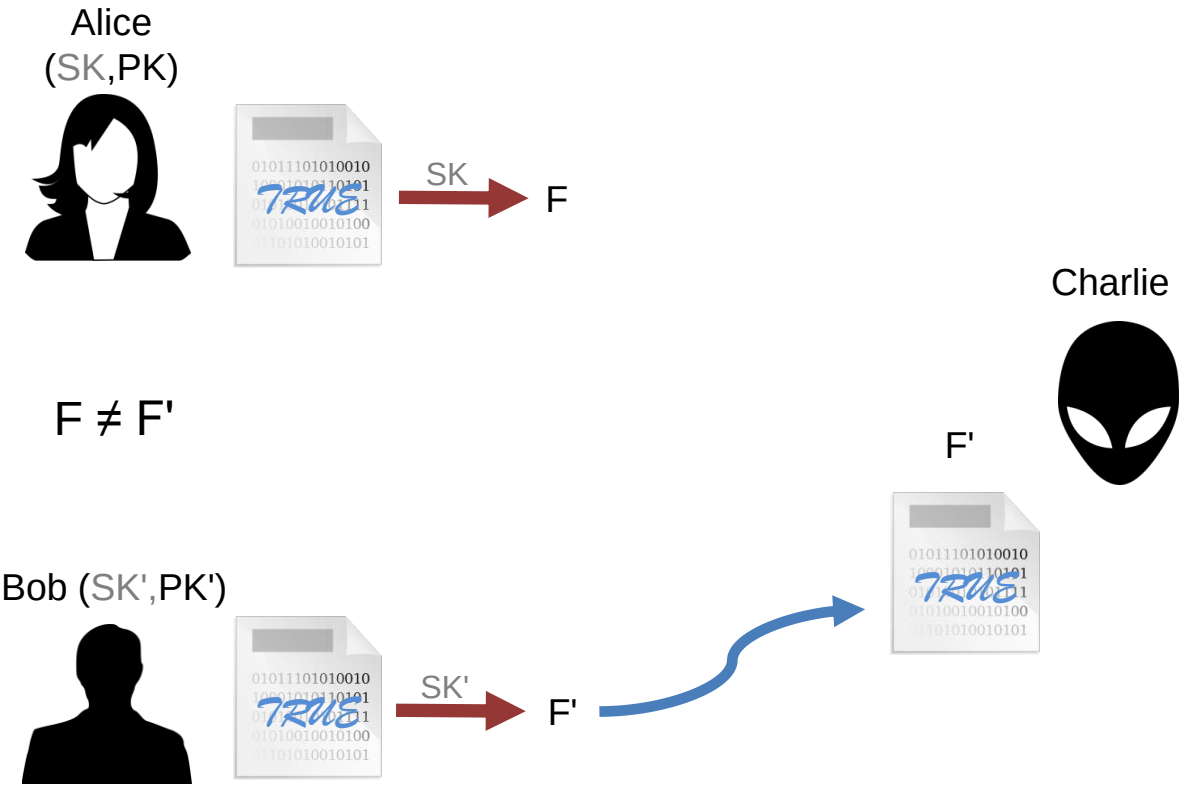
F ≠ F'

Bob (SK',PK')

SK' → F'

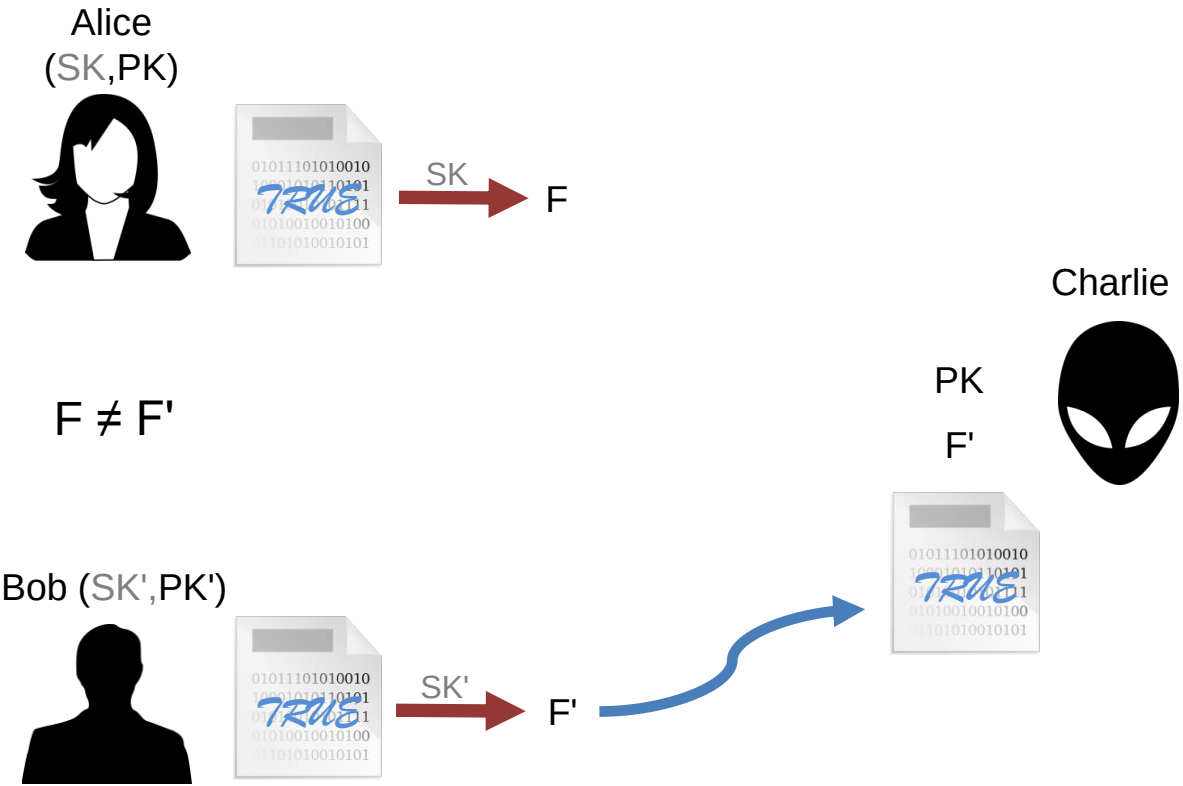
Properties of Digital signatures

We can sign only with SK



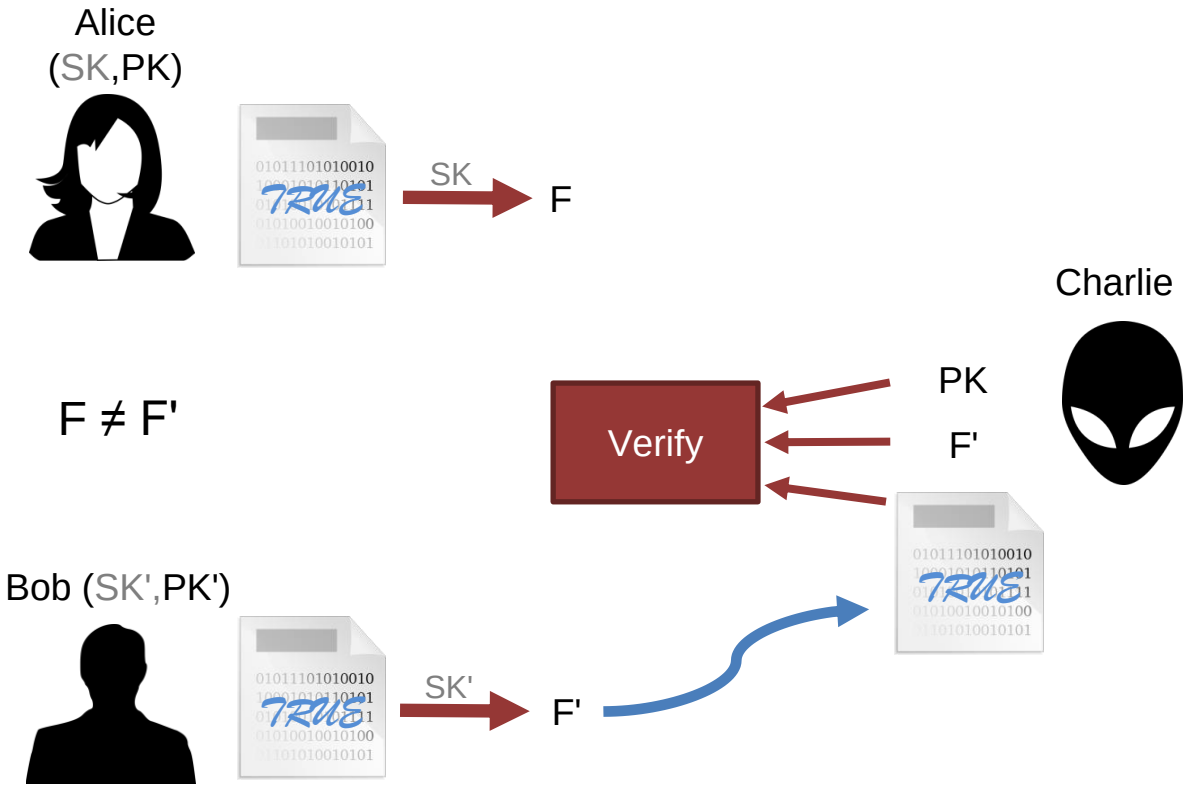
Properties of Digital signatures

We can sign only with SK



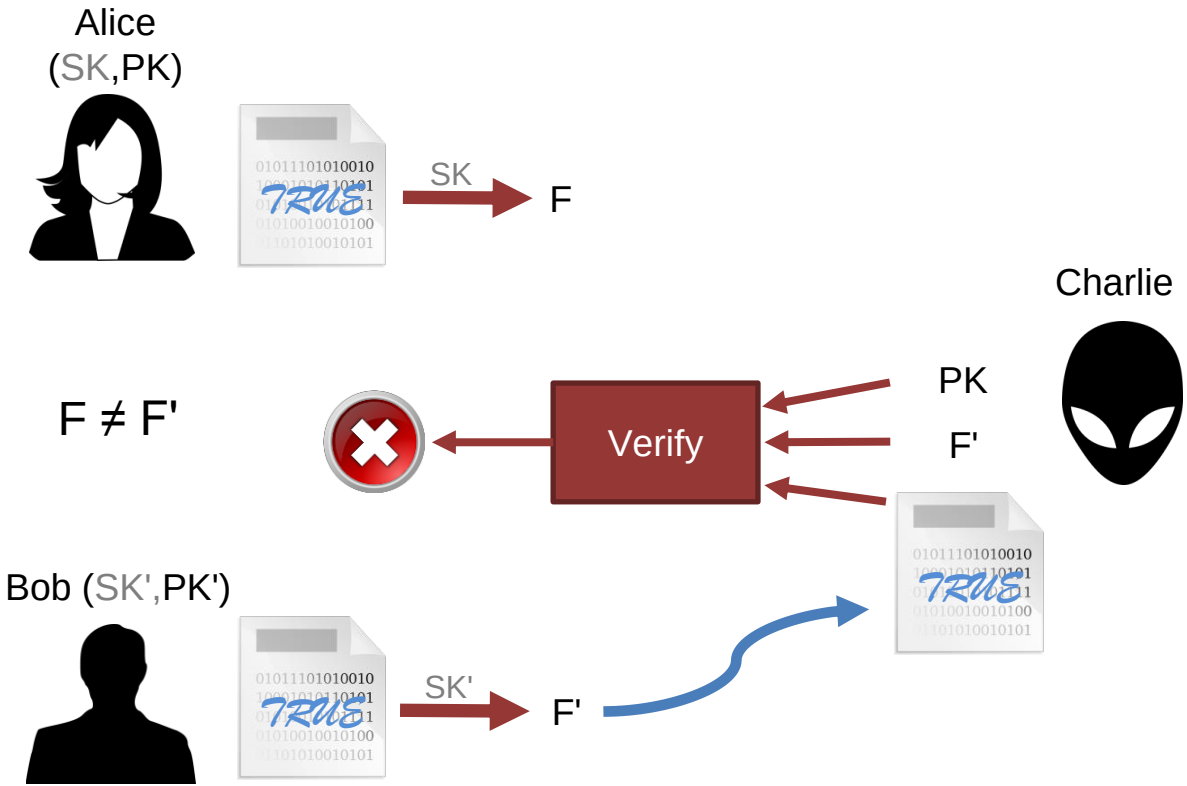
Properties of Digital signatures

We can sign only with SK



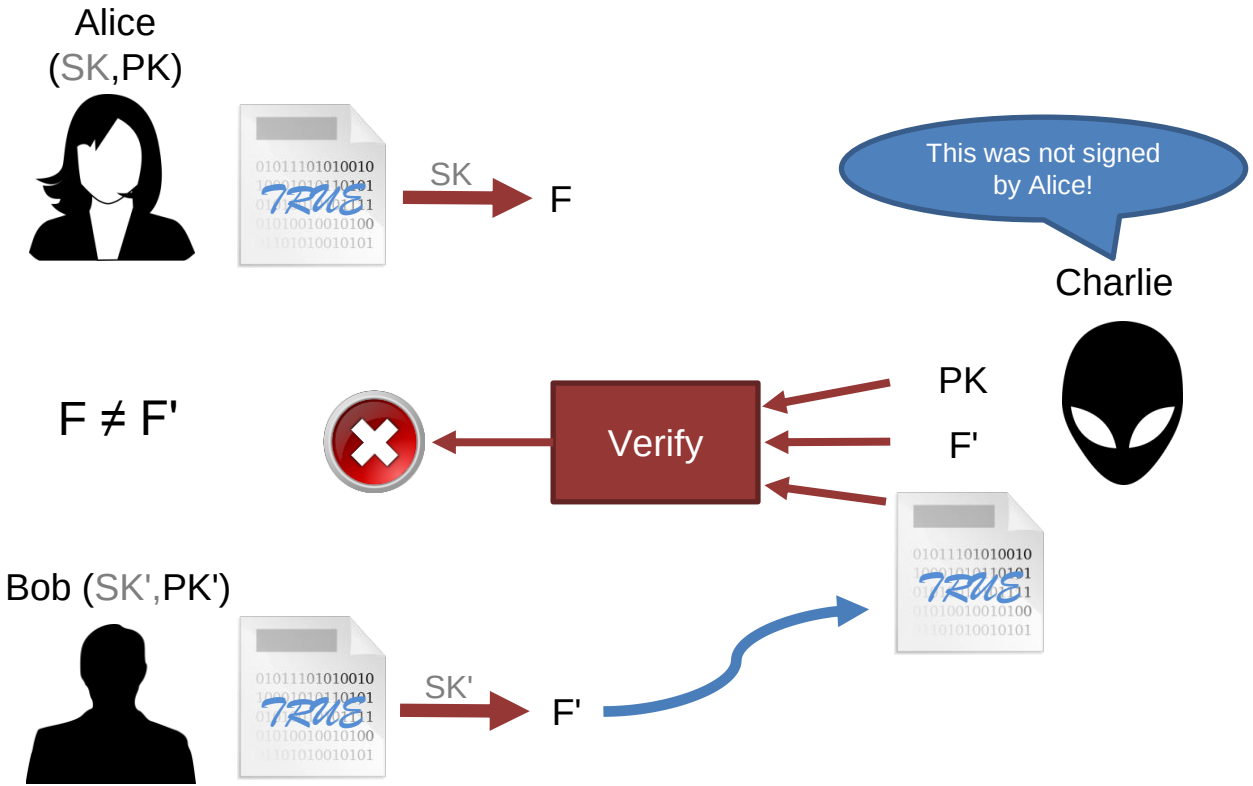
Properties of Digital signatures

We can sign only with SK



Properties of Digital signatures

We can sign only with SK



Problem 2 of e-Kuna

Who adds the transactions?

Alice

Ledger

Alice pays Bob $50

Bob

Charlie

Problem 2 of e-Kuna

Who adds the transactions?

Alice

Ledger

Alice pays Bob $50 F-A

Bob

Charlie

Problem 2 of e-Kuna

Who adds the transactions?

Alice

Ledger

Alice pays Bob $50 F-A

Alice payed me $50!

Bob

Charlie

Problem 2 of e-Kuna

Who adds the transactions?

Alice

Ledger

Alice pays Bob $50 F-A

Alice pays Bob $1000

Bob

Charlie

Problem 2 of e-Kuna

Who adds the transactions?

Alice

Charlie

Ledger

Alice pays Bob $50 F-A

Alice pays Bob $1000 ???

Bob

Problem 2 of e-Kuna

Who adds the transactions?

Alice

Ledger

Alice pays Bob $50 F-A

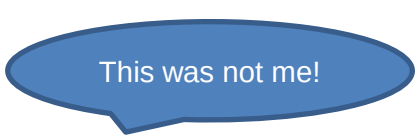
Alice pays Bob $1000 F-B

Bob

Charlie

Problem 2 of e-Kuna

Who adds the transactions?



Alice

Ledger

Alice pays Bob $50 F-A

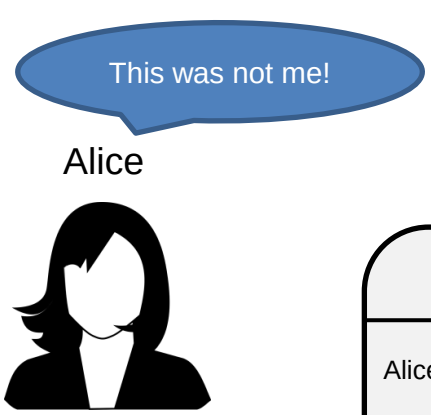
Alice pays Bob $1000 F-B

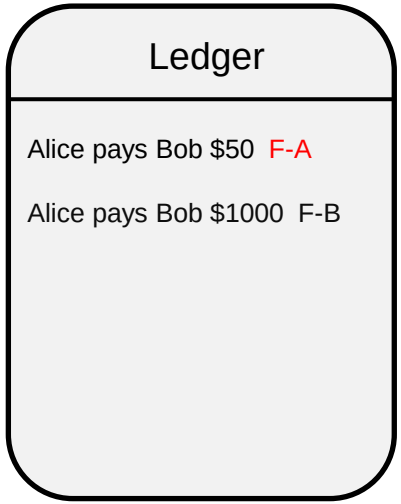
Bob

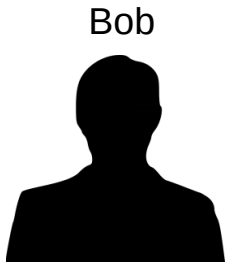
Charlie

Problem 2 of e-Kuna

Who adds the transactions?







Practical considerations

Randomness:

- We need a good source of randomness
- To generate the keys
- To generate the signature (each signature!!!)

Practical considerations

Size of the signature:

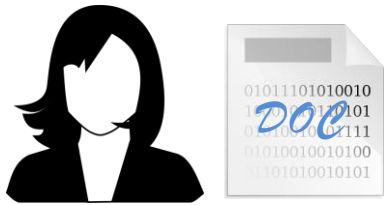
- Classical algorithms depend on the size of the document
- Solution used these days is to sign the hash of the document

Practical considerations

Size of the signature:

- Classical algorithms depend on the size of the document
- Solution used these days is to sign the hash of the document

Alice (SK,PK)

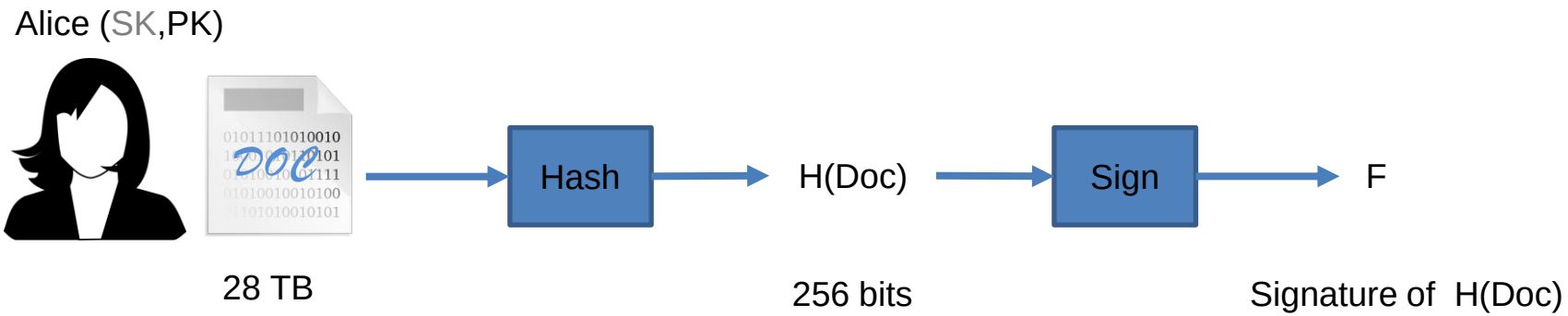


28 TB

Practical considerations

Size of the signature:

- Classical algorithms depend on the size of the document
- Solution used these days is to sign the hash of the document



Digital signature in BitCoin

Elliptic curve digital signature algorithm (ECDSA)

- NIST/NSA standard
- Elliptic curve secp256k1
- 2^{128} bits of security (number of operations needed to break the encryption)

Important sizes:

- SK = 256 bits
- PK = 512 bits (257 bits compressed)
- Message size = 256 bits (well, that's convenient)
- Signature size = 512 bits

BitCoin users

PK allows:

- Link a signature to an entity (the person controlling the associated SK)

BitCoin users

PK allows:

- Link a signature to an entity (the person controlling the associated SK)

User in a decentralized system:

- A PK (that is, a person controlling the SK for this PK)

Peculiarities

Of users in a decentralized system

What is I need more than one user?

- Run the key generation algorithm again
- You can have any number of users (can be problematic)

Is it safe?

- Since no one controls the PKs and SKs, can someone generate mine?
- In theory yes, but practically, the probability is virtually 0 (given a good source of randomness)

Peculiarities

Of users in a decentralized system

Is it safe? (2)

- 100%
- If you manage well your SK (and the signatures – randomness again)

What happens if I lose my SK?

- 0% possibility to recuperate it
- Big benefit of banks is allowing to recuperate your password or credit card when lost
- BitCoin is 100% secure cryptographically, but it does not allow this!