Transacciones y contratos inteligentes

Timelocks 1


```
{
    "txid": "7cf0883a4a24664a1c77fdd465782f0d7c962b7942daa4408f3083ea659aef6a",
    "version": 2,
    "locktime": 500114,
    "vin": [
        {
            "txid": "42a3fdd7d7baea12221f259f38549930b47cec288b55e4a8facc3c899f4775da",
            "vout": 0,
            "scriptSig": "473044022048d1468895910edafe53d4ec4209192cc3a8f0f21e7b9811f83b5e419bfb57
            e002203fef249b56682dbbb1528d4338969abb14583858488a3a766f609185efe68bca0121031a455dab5e
            1f614e574a2f4f12f22990717e93899695fb0d81e4ac2dcfd25d00",
            "sequence": 4294967295
        }
    ],
    "vout": [
        {
            "value": 0.00990000,
            "n": 0,
            "scriptPubKey": "OP_HASH160 e9c3dd0c07aac76179ebc76a6c78d4d67c6c160a OP_EQUAL",

        }
    ]
}
```

Timelocks 1

locktime:

- 0 significa propagar la transacción inmediatamente
- 0 < x < 500 000 000 – transacción no válida hasta el bloque x
- x > 500 000 000 – transacción no válida hasta unix epoch time x

- Un lock de la transacción entera!!!

Timelocks 1

Aquí está mi transacción. La puedes gastar en 50 días ☺

Alice

Bob


```
{
    "txid": "7cf0883a4a24664a1c77fdd465782f0d7c962b7942daa4408f3083ea659aef6a",
    "version": 2,
    "locktime": 500114,
    "vin": [
        {
            "txid": "42a3fdd7d7baea12221f259f38549930b47cec288b55e4a8facc3c899f4775da",
            "vout": 0,
            "scriptSig": "473044022048d1468895910edafe53d4ec4209192cc3a8f0f21e7b9811f83b5e419bfb57
            e002203fef249b56682dbbb1528d4338969abb14583858488a3a766f609185efe68bca0121031a455dab5e
            1f614e574a2f4f12f22990717e93899695fb0d81e4ac2dcfd25d00",
            "sequence": 4294967295
        }
    ],
    "vout": [
        {
            "value": 0.00990000,
            "n": 0,
            "scriptPubKey": "OP_HASH160 e9c3dd0c07aac76179ebc76a6c78d4d67c6c160a OP_EQUAL",

        }
    ]
}
```


Bob

Timelocks 1

Sabes que, me voy a gastar esta plata.

Alice

Bob


```
{
    "txid": "7cf0883a4a24664a1c77fdd465782f0d7c962b7942daa4408f3083ea659aef6a",
    "version": 2,
    "locktime": 500114,
    "vin": [
        {
            "txid": "42a3fdd7d7baea12221f259f38549930b47cec288b55e4a8facc3c899f4775da",
            "vout": 0,
            "scriptSig": "473044022048d1468895910edafe53d4ec4209192cc3a8f0f21e7b9811f83b5e419bfb57
            e002203fef249b56682dbbb1528d433
            1f614e574a2f4f12f22990717e93899
            "sequence": 4294967295
        }
    ],
    "vout": [
        {
            "value": 0.00990000,
            "n": 0,
            "scriptPubKey": "OP_HASH160 e9c
        }
    ]
}
```

```
{
    "version": 1,
    "locktime": 0,
    "vin": [
        {
            "txid": "42a3fdd7d7baea12221f259f38549930b47cec288b55e4a8facc3c899f4775d
            "vout": 0,
            "scriptSig": "304402207
            0f45e2590602201b06ecbbbf6fae0a3455d98ac2924cd6dc3022425a2f08c60ffb4
            e9[ALL] 04a84d304aa8963fbd36287e674f109827b6d6ea60d57a7d9357df03be
            475ca128b1a50408b7f584041ffd52d6b19aba5256e99dcdbbe2ed7373775d"
            6dbdc8
            cedb2c47
        },
        "sequence": 4294967295
    ],
    "vout": [
        {
            "value": 0.10000000,
            "n": 0,
            "scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHECKSIG"
        }
    ]
}
```


Alice

Timelocks 2 (CLTV)

Para evitar esto BIP 65 introduce nueva instrucción:

- OP_CHECKLOCKTIMEVERIFY (OP_CLTV)
- Ahora uno puede usar locktime en el nivel de UTXOs

CLTV

Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	4	012123…
Outputs			
num	value	Locking script	
0	3.1	OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG	

CLTV

Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	4	012123…

Outputs		
num	value	Locking script
0	3.1	

<now + 30 days> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

CLTV

UTXO(Tx: 0xff, Nr_out: 0):

<now + 30 days> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Bob

CLTV

UTXO(Tx: 0xff, Nr_out: 0):

<now + 30 days> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Bob

Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	0	
Outputs			
num	value	Locking script	
0	3.1	OP_DUP OP_HASH...	

CLTV

UTXO(Tx: 0xff, Nr_out: 0):

<now + 30 days> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Bob

Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	0	<sig Bob> <PK Bob>
Outputs			
num	value	Locking script	
0	3.1	OP_DUP OP_HASH...	

CLTV

UTXO(Tx: 0xff, Nr_out: 0):

<now + 30 days> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Bob

Locktime: <now + 100 days>			
Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	0	<sig Bob> <PK Bob>
Outputs			
num	value	Locking script	
0	3.1	OP_DUP OP_HASH...	

CLTV validación

UTXO(Tx: 0xff, Nr_out: 0):

<now + 30 days> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Locktime: <now + 100 days>			
Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	0	<sig Bob> <PK Bob>

CLTV validación

UTXO(Tx: 0xff, Nr_out: 0):

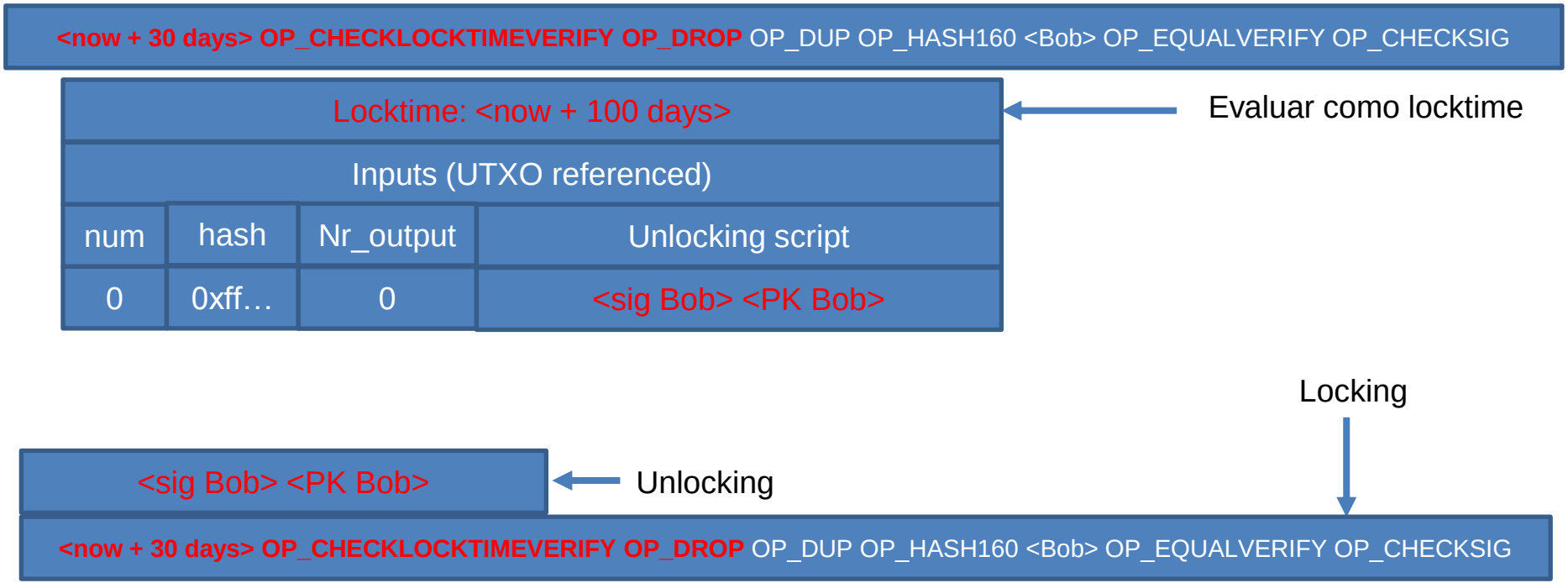
<now + 30 days> OP_CHECKLOCKTIMEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Locktime: <now + 100 days>			
Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	0	<sig Bob> <PK Bob>

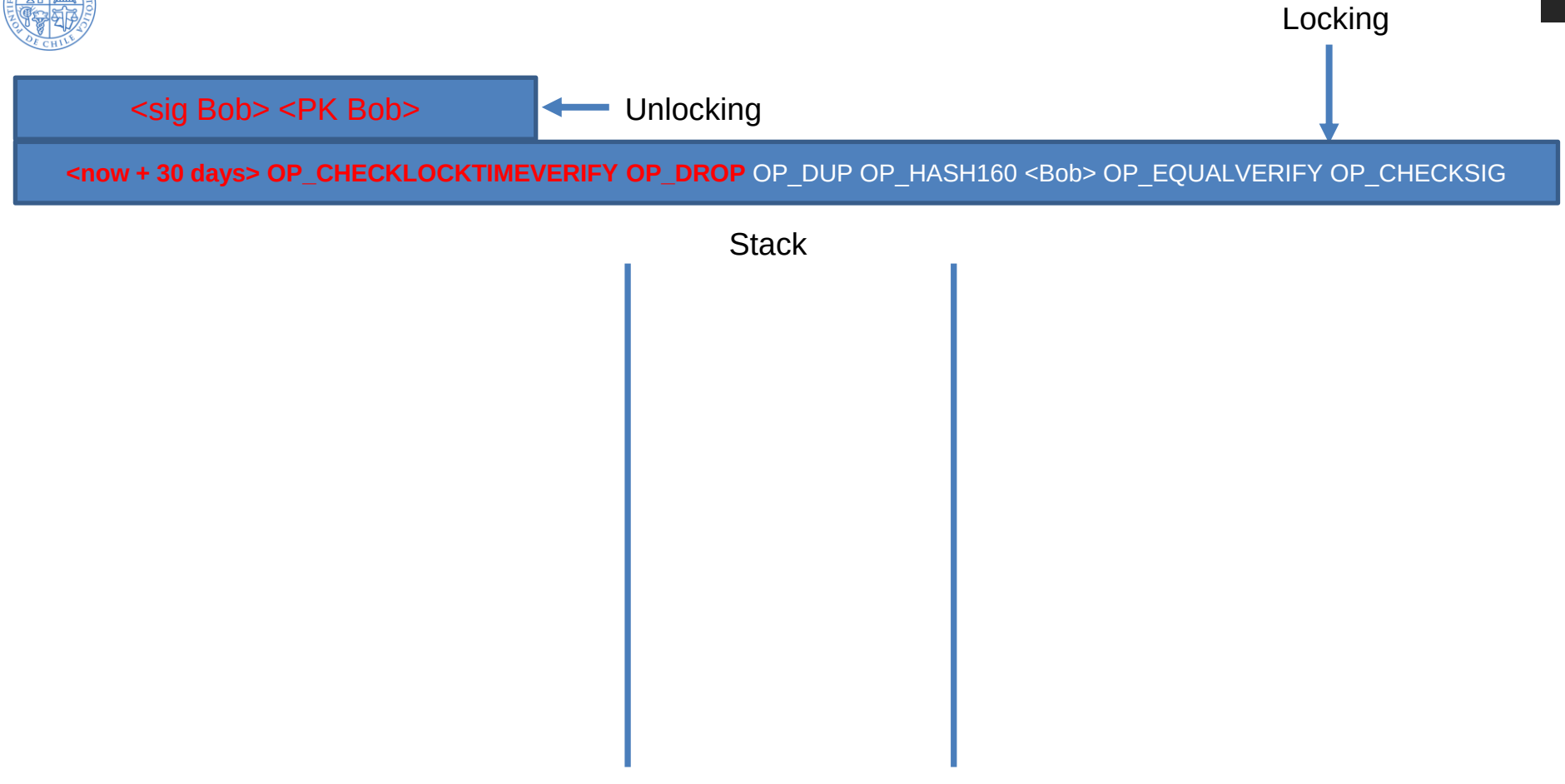
Evaluar como locktime

CLTV validación

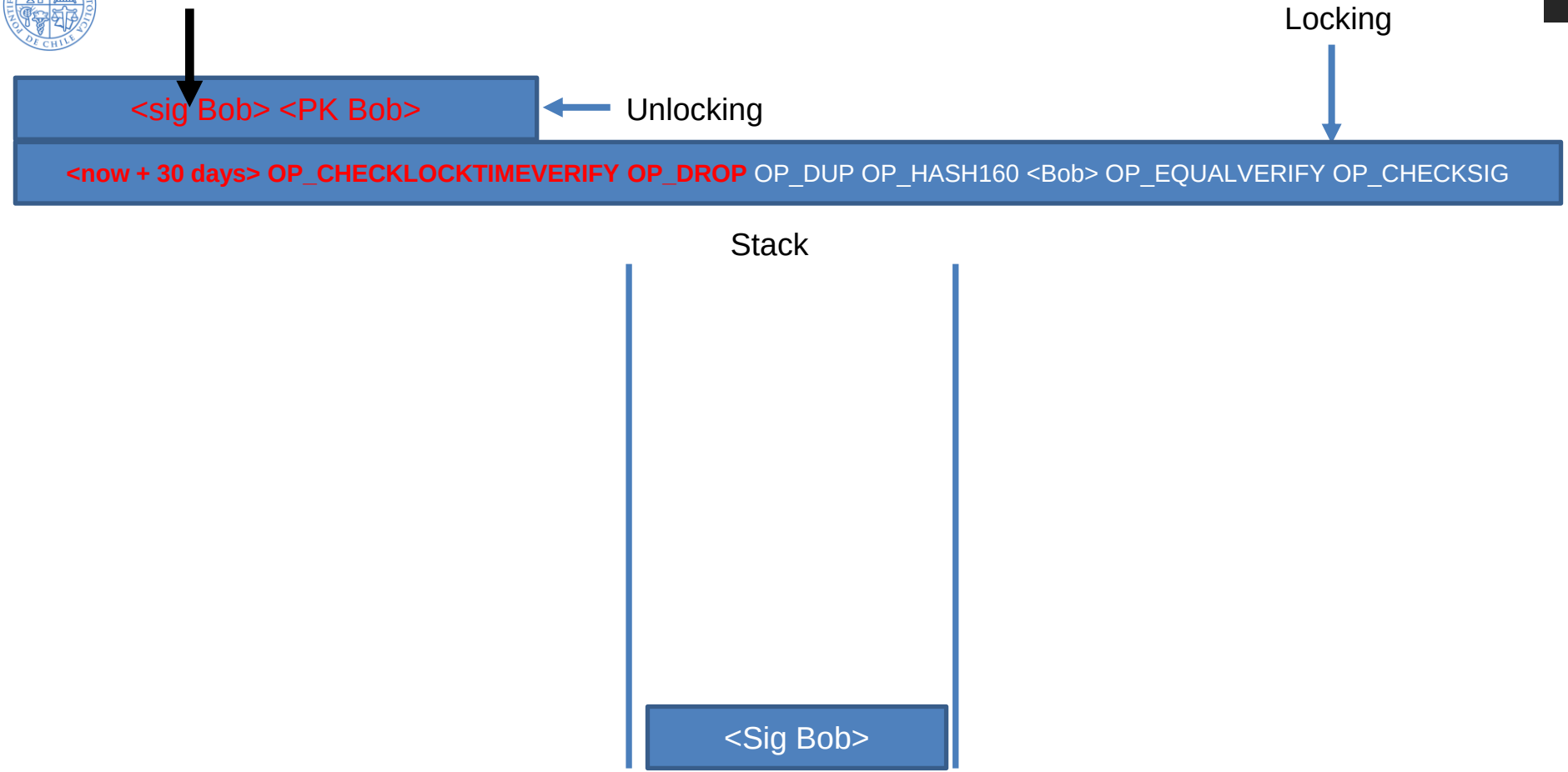
UTXO(Tx: 0xff, Nr_out: 0):



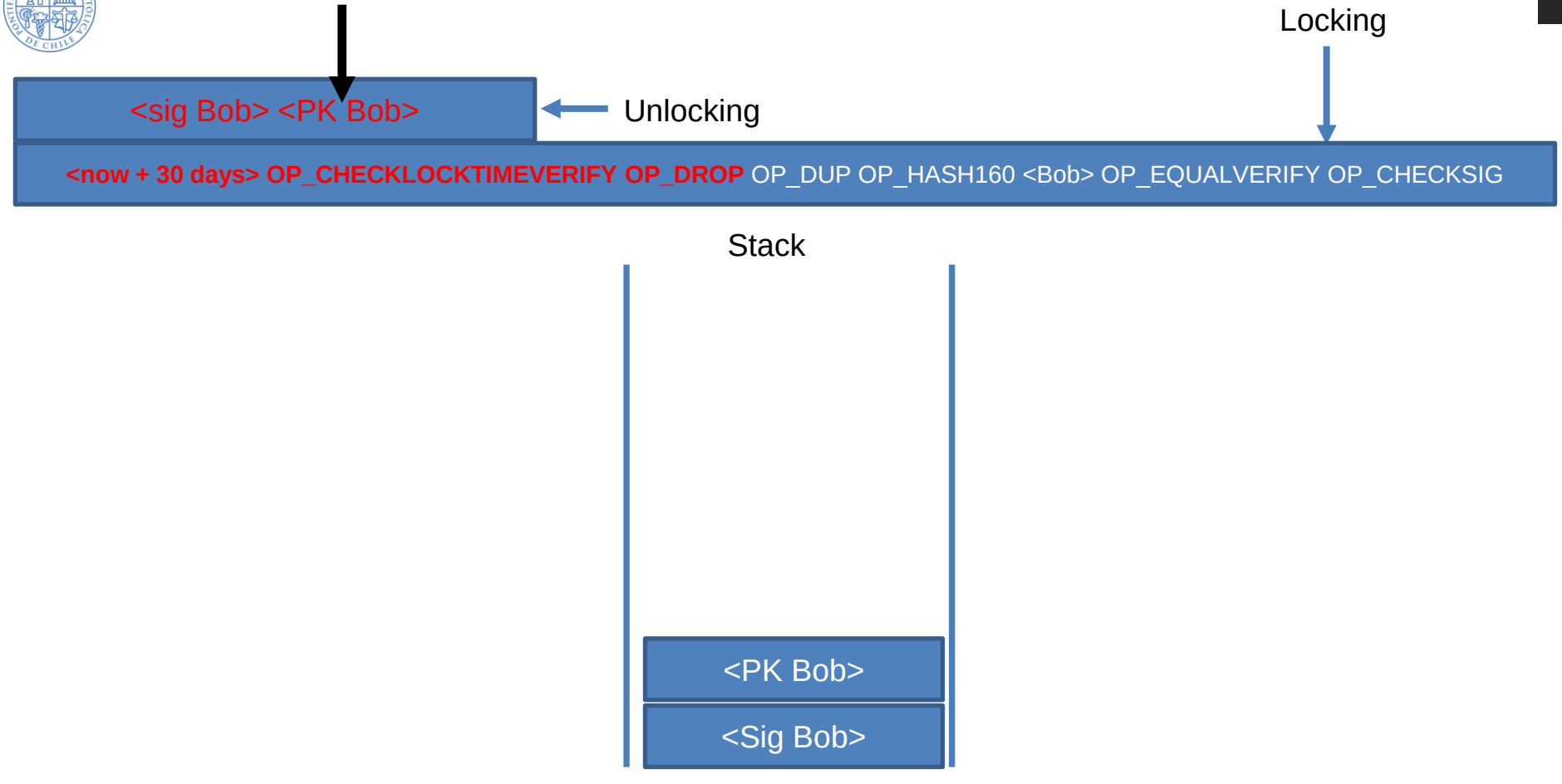
CLTV validación



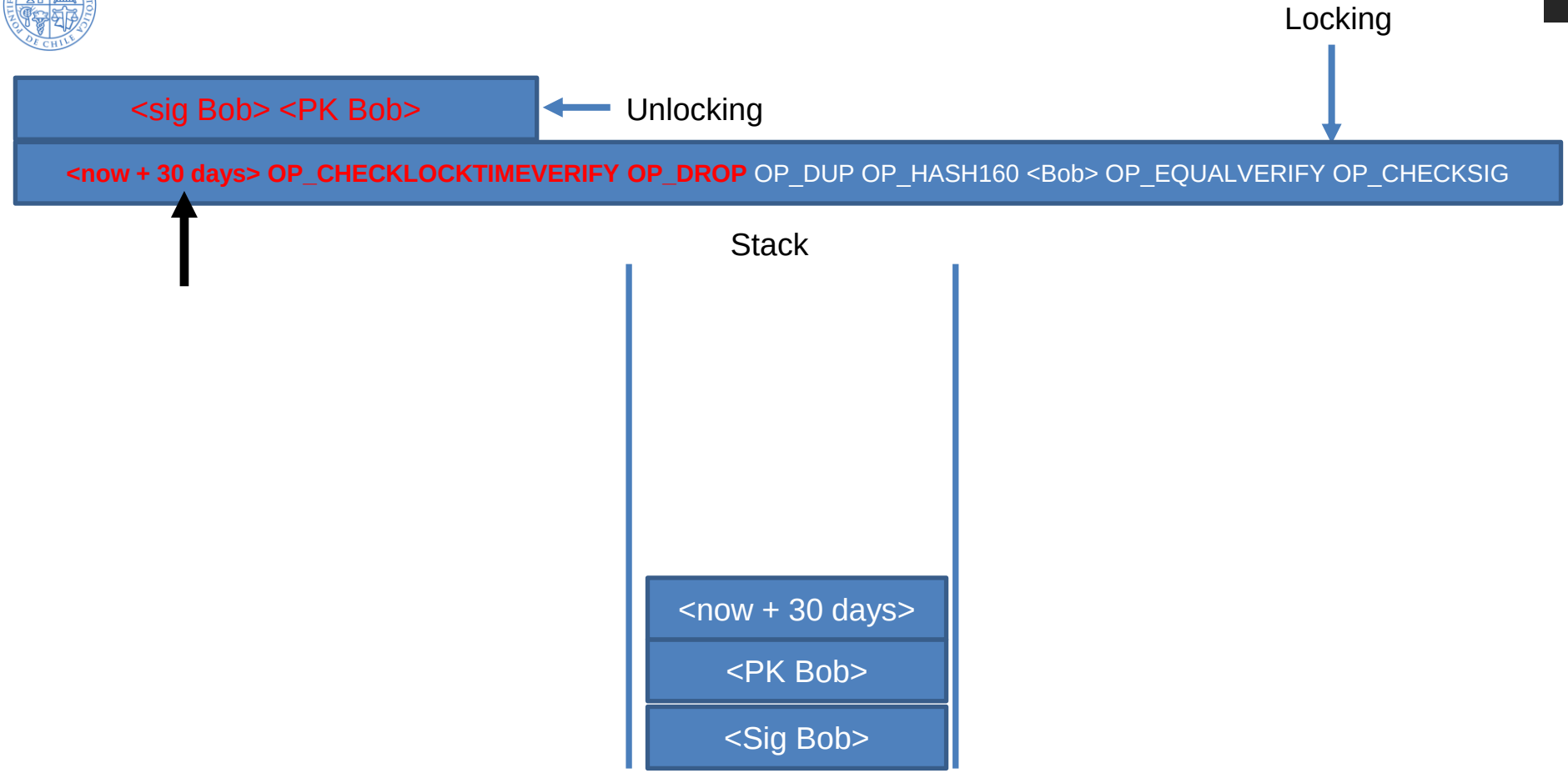
CLTV validación



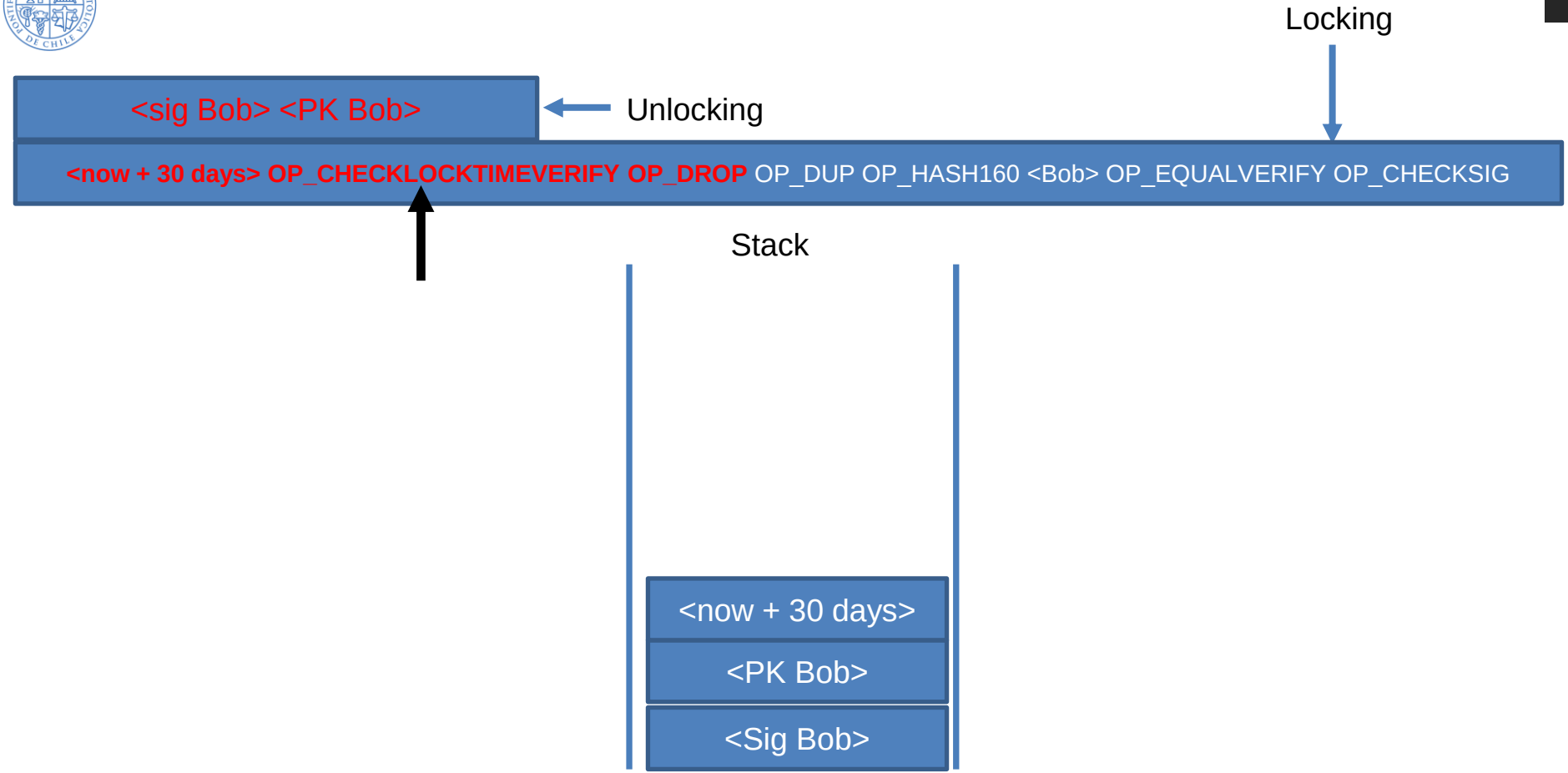
CLTV validación



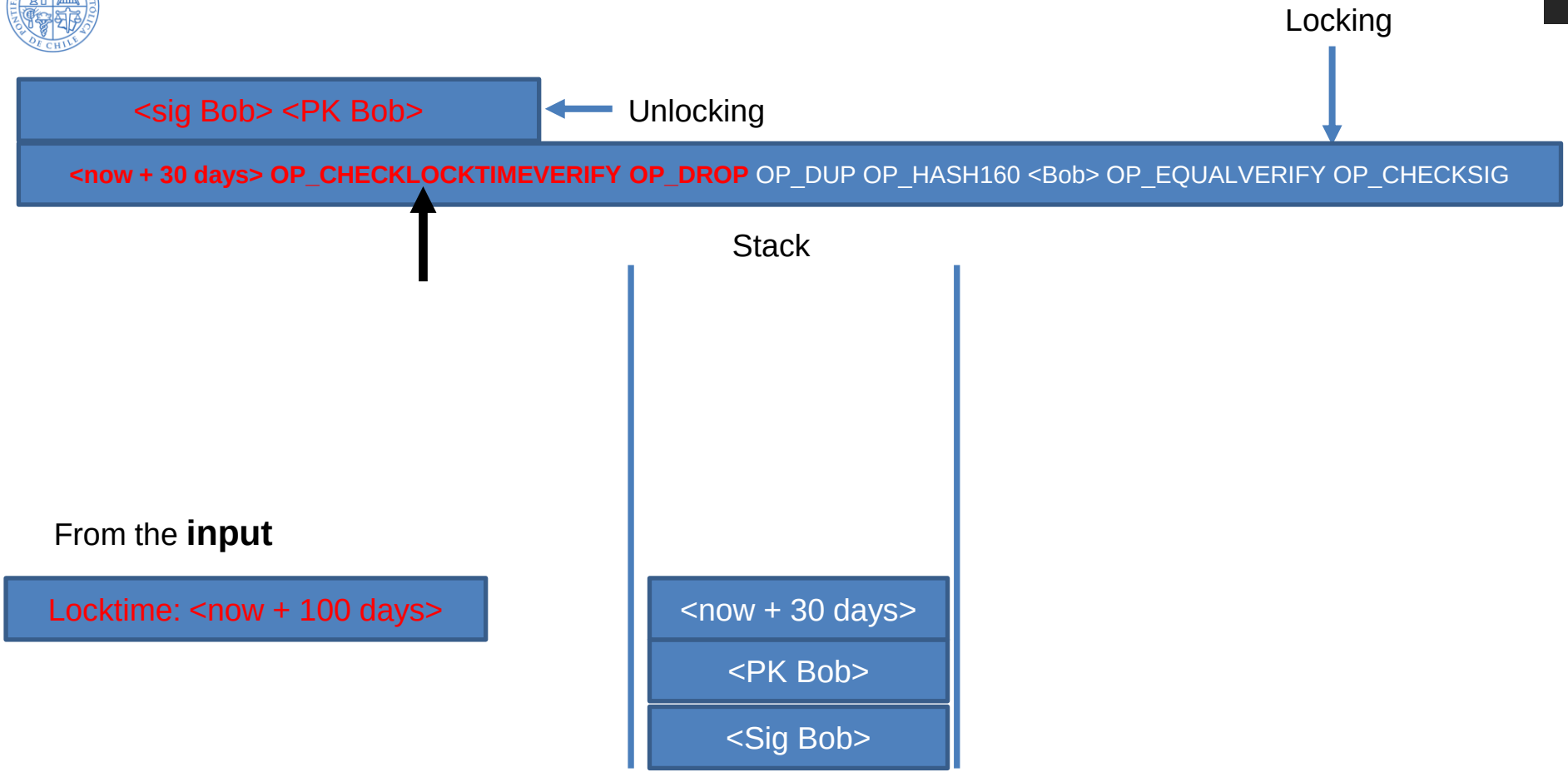
CLTV validación



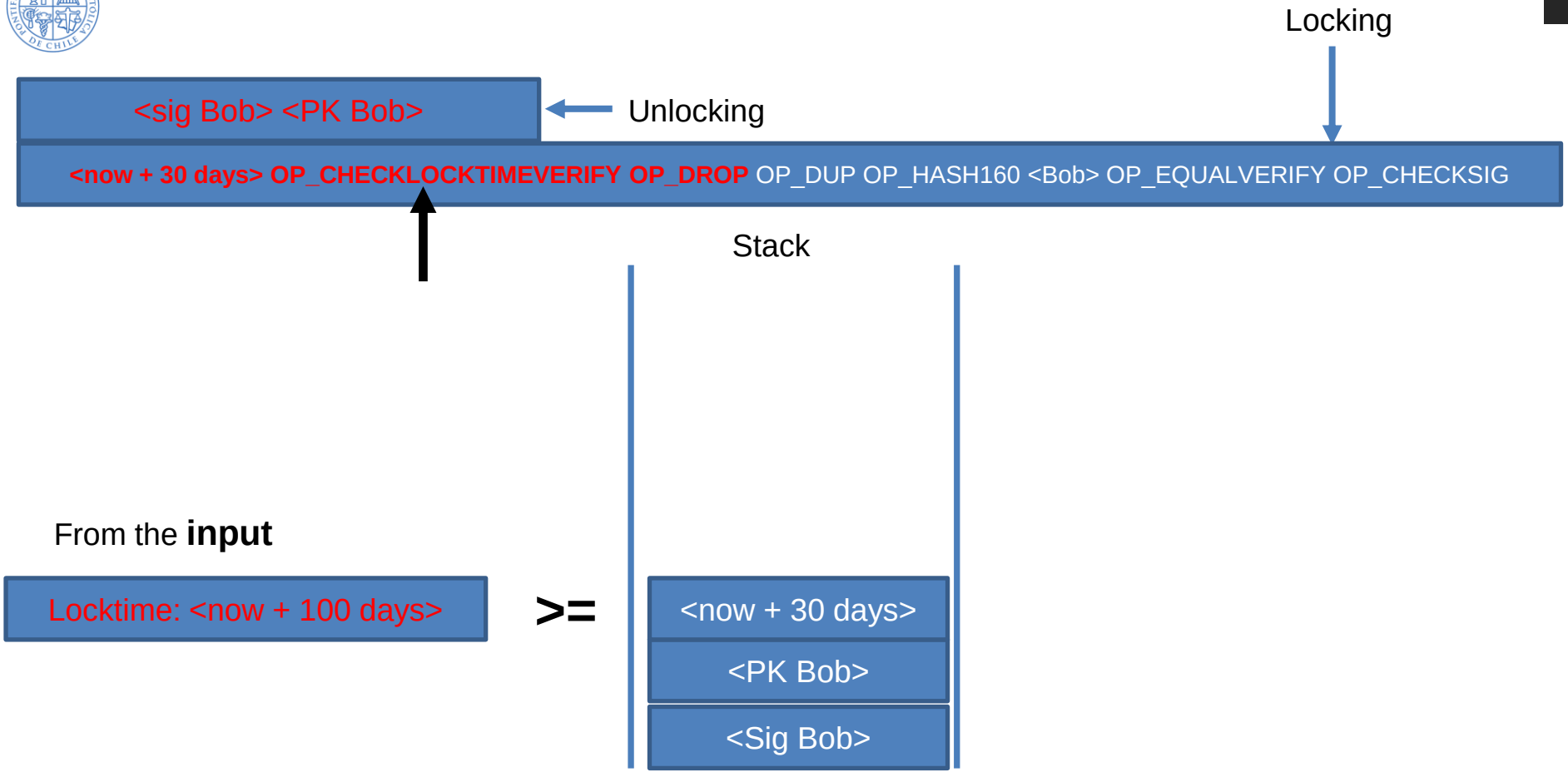
CLTV validación



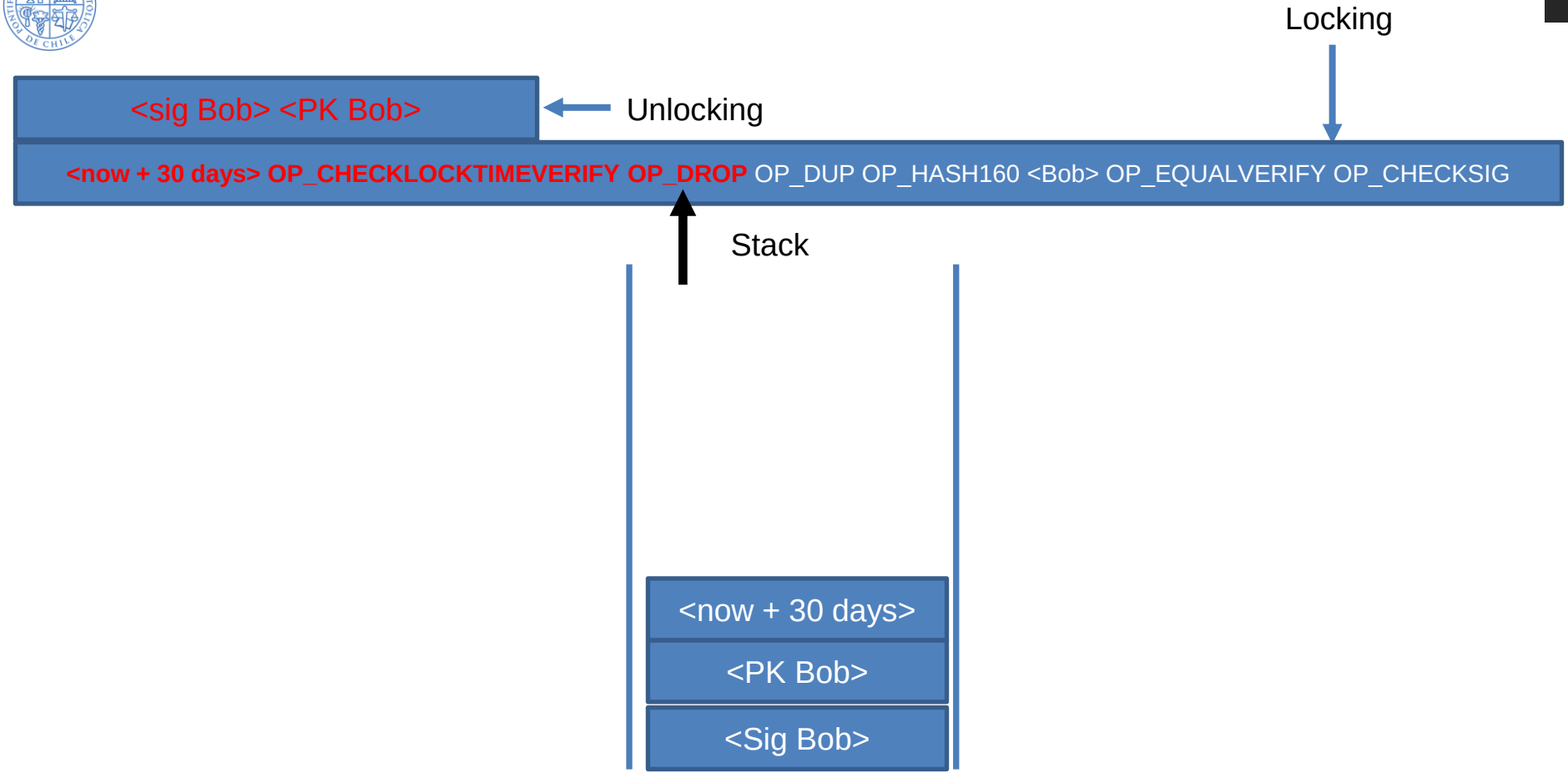
CLTV validación



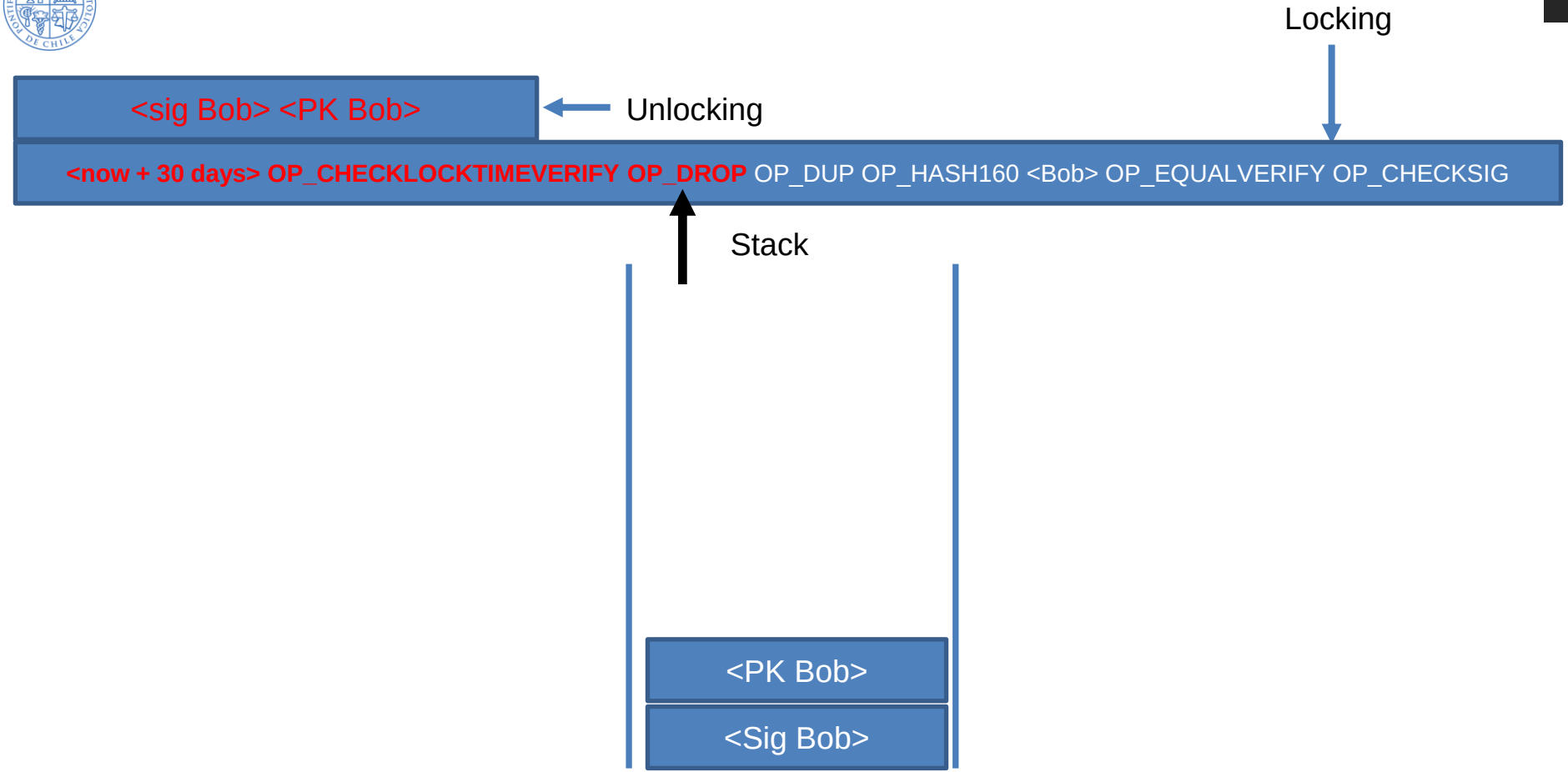
CLTV validación



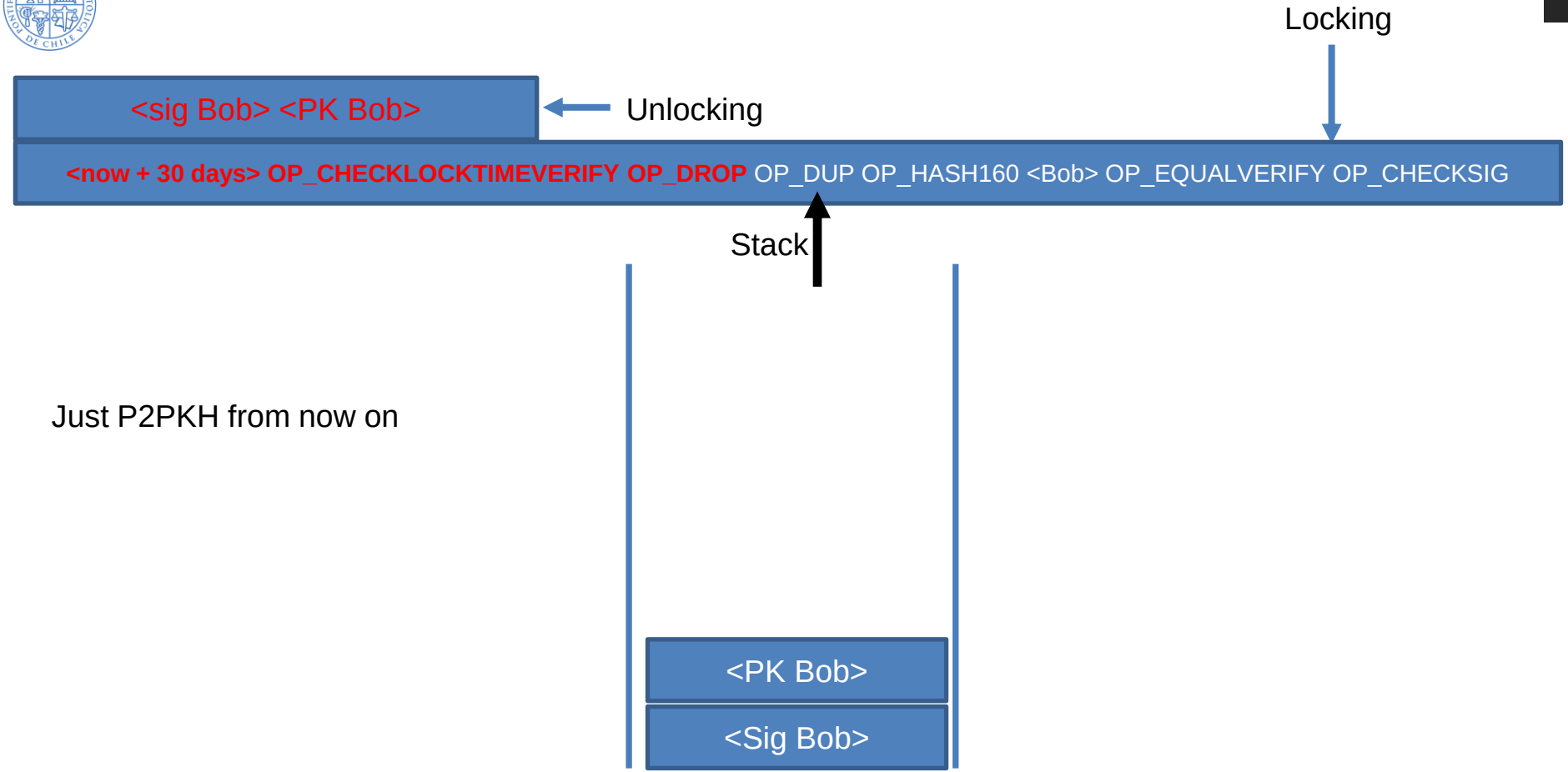
CLTV validación



CLTV validación



CLTV validación



Just P2PKH from now on

Timelocks 3


```
{
    "txid": "7cf0883a4a24664a1c77fdd465782f0d7c962b7942daa4408f3083ea659aef6a",
    "version": 2,
    "locktime": 500114,
    "vin": [
        {
            "txid": "42a3fdd7d7baea12221f259f38549930b47cec288b55e4a8facc3c899f4775da",
            "vout": 0,
            "scriptSig": "473044022048d1468895910edafe53d4ec4209192cc3a8f0f21e7b9811f83b5e419bfb57
            e002203fef249b56682dbbb1528d4338969abb14583858488a3a766f609185efe68bca0121031a455dab5e
            1f614e574a2f4f12f22990717e93899695fb0d81e4ac2dcfd25d00",
            "sequence": 4
        }
    ],
    "vout": [
        {
            "value": 0.00990000,
            "n": 0,
            "scriptPubKey": "OP_HASH160 e9c3dd0c07aac76179ebc76a6c78d4d67c6c160a OP_EQUAL",

        }
    ]
}
```

Timelocks 3 (CSV)

Para tener timelock relativo BIP112 introduce:

- OP_CHECKSEQUENCEVERIFY (OP_CSV)
- Ahora uno puede usar locktime en el nivel de UTXOs/Input
- Uso del parámetro sequence

- Uno para cada input

Timelocks 3

Solo se puede gastar si el UTXO tiene 4 confirmaciones


```
{
    "txid": "7cf0883a4a24664a1c77fdd465782f0d7c962b7942daa4408f3083ea659aef6a",
    "version": 2,
    "locktime": 500114,
    "vin": [
        {
            "txid": "42a3fdd7d7baea12221f259f38549930b47cec288b55e4a8facc3c899f4775da",
            "vout": 0,
            "scriptSig": "473044022048d1468895910edafe53d4ec4209192cc3a8f0f21e7b9811f83b5e419bfb57
            e002203fef249b56682dbbb1528d4338969abb14583858488a3a766f609185efe68bca0121031a455dab5e
            1f614e574a2f4f12f22990717e93899695fb0d81e4ac2dcfd25d00",
            "sequence": 4
        }
    ],
    "vout": [
        {
            "value": 0.00990000,
            "n": 0,
            "scriptPubKey": "OP_HASH160 e9c3dd0c07aac76179ebc76a6c78d4d67c6c160a OP_EQUAL",

        }
    ]
}
```

Timelocks 3


```
{
    "txid": "7cf0883a4a24664a1c77fdd465782f0d7c962b7942daa4408f3083ea659aef6a",
    "version": 2,
    "locktime": 500114,
    "vin": [
        {
            "txid": "42a3fdd7d7baea12221f259f38549930b47cec288b55e4a8facc3c899f4775da",
            "vout": 0,
            "scriptSig": "473044022048d1468895910edafe53d4ec4209192cc3a8f0f21e7b9811f83b5e419bfb57
            e002203fef249b56682dbbb1528d4338969abb14583858488a3a766f609185efe68bca0121031a455dab5e
            1f614e574a2f4f12f22990717e93899695fb0d81e4ac2dcfd25d00",
            "sequence": 4
        }
    ],
    "vout": [
        {
            "value": 0.00990000,
            "n": 0,
            "scriptPubKey": "OP_HASH160 e9c3dd0c07aac76179ebc76a6c78d4d67c6c160a OP_EQUAL",

        }
    ]
}
```


Si no, la transacción no está válida

Solo se puede gastar si el UTXO tiene 4 confirmaciones

CSV

Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	4	012123…

Outputs		
num	value	Locking script
0	3.1	

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

CSV

UTXO(Tx: 0xff, Nr_out: 0):

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Bob

Inputs (UTXO referenced)			
num	hash	Nr_output	Unlocking script
0	0xff…	0	<sig Bob> <PK Bob>
Outputs			
num	value	Locking script	
0	3.1	OP_DUP OP_HASH...	

CSV

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

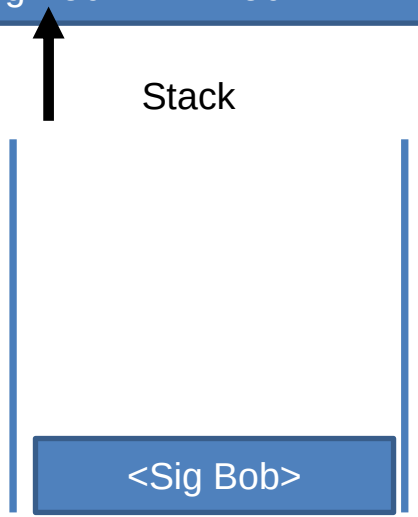
Valido?

Stack

CSV

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

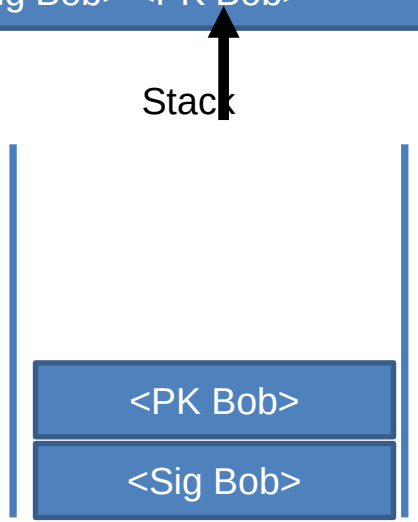
Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41



CSV

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

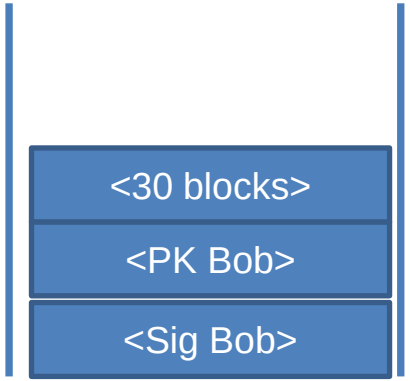


CSV

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

Stack

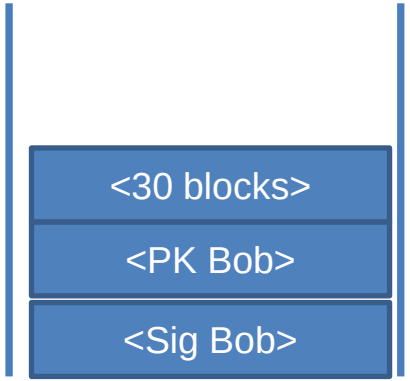


CSV

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

Stack



CSV

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

Stack

From the **input**

Sequence: 41

<30 blocks>

<PK Bob>

<Sig Bob>

CSV

<30 blocks> OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160 <Bob> OP_EQUALVERIFY OP_CHECKSIG

Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

Stack

From the **input**

Sequence: 41

>=

<30 blocks>

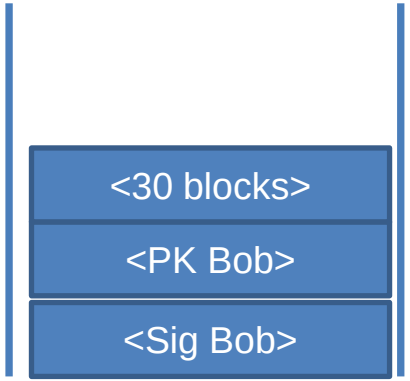
<PK Bob>

<Sig Bob>

CSV

Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

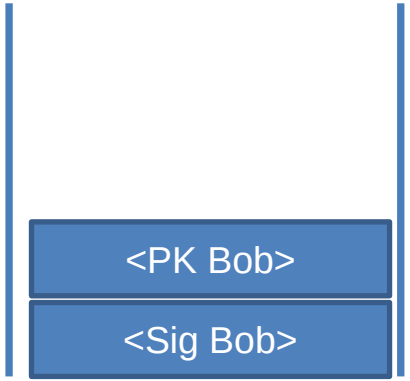
Stack



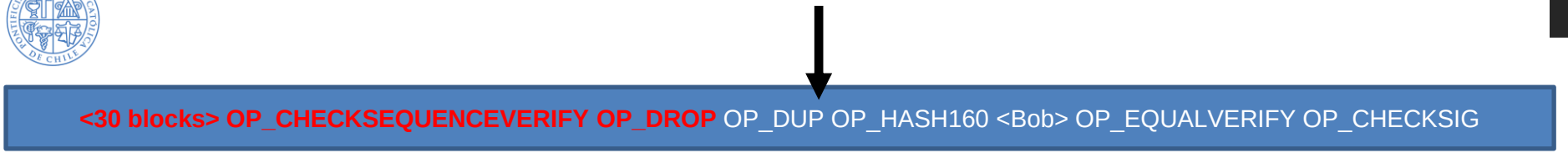
CSV

Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

Stack



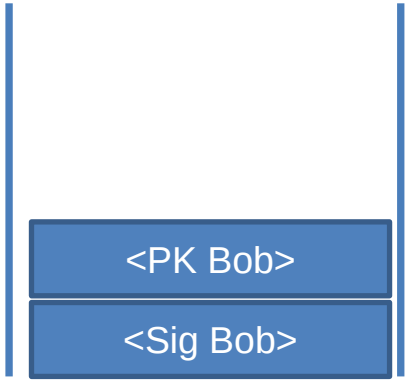
CSV



Inputs (UTXO referenced)				
num	hash	Nr_output	Unlocking script	sequence
0	0xff…	0	<sig Bob> <PK Bob>	41

P2PKH standard

Stack



IF ELSE

Para cosas interesantes

Usual code:

```
if (condition):
    code to run when true
else:
    code to run when false
code to run always
```

With stack:

```
(condition)
IF
    code to run when true
ELSE
    code to run when false
ENDIF
code to run always
```

IF ELSE

1-2 MULTISIG

Locking script:

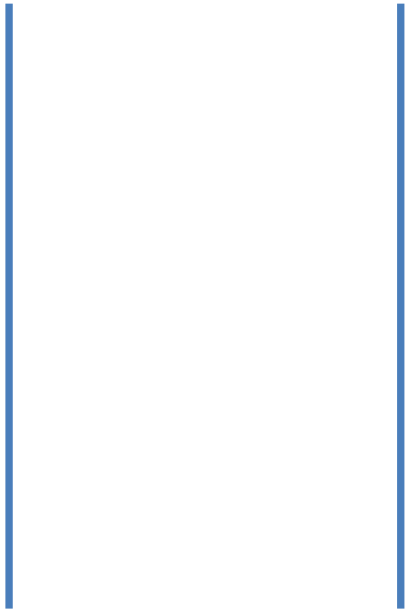
```
IF
    <PK Alice> OP_CHECKSIG
ELSE
    <PK Bob> OP_CHECKSIG
ENDIF
```

Como activar IF/ELSE?

Multisig

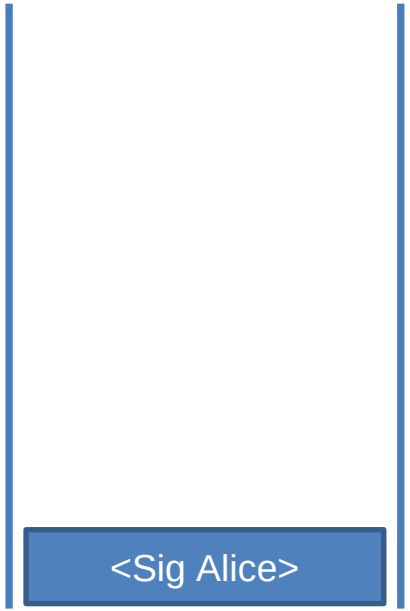
<sig Alice> 1	IF <PK Alice> OP_CHECKSIG ELSE <PK Bob> OP_CHECKSIG

Stack



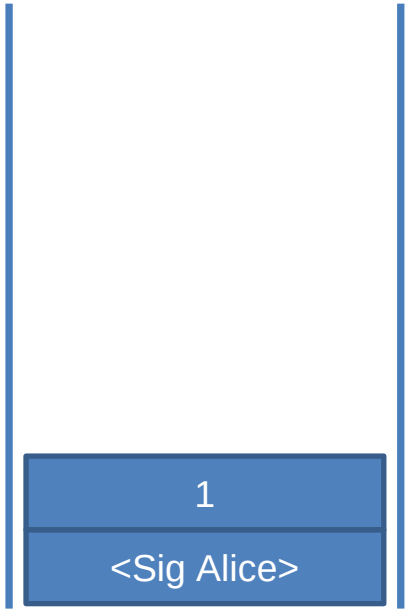
Multisig

Stack



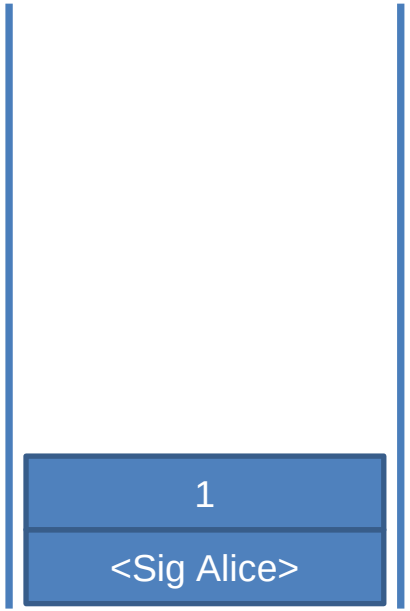
Multisig

Stack

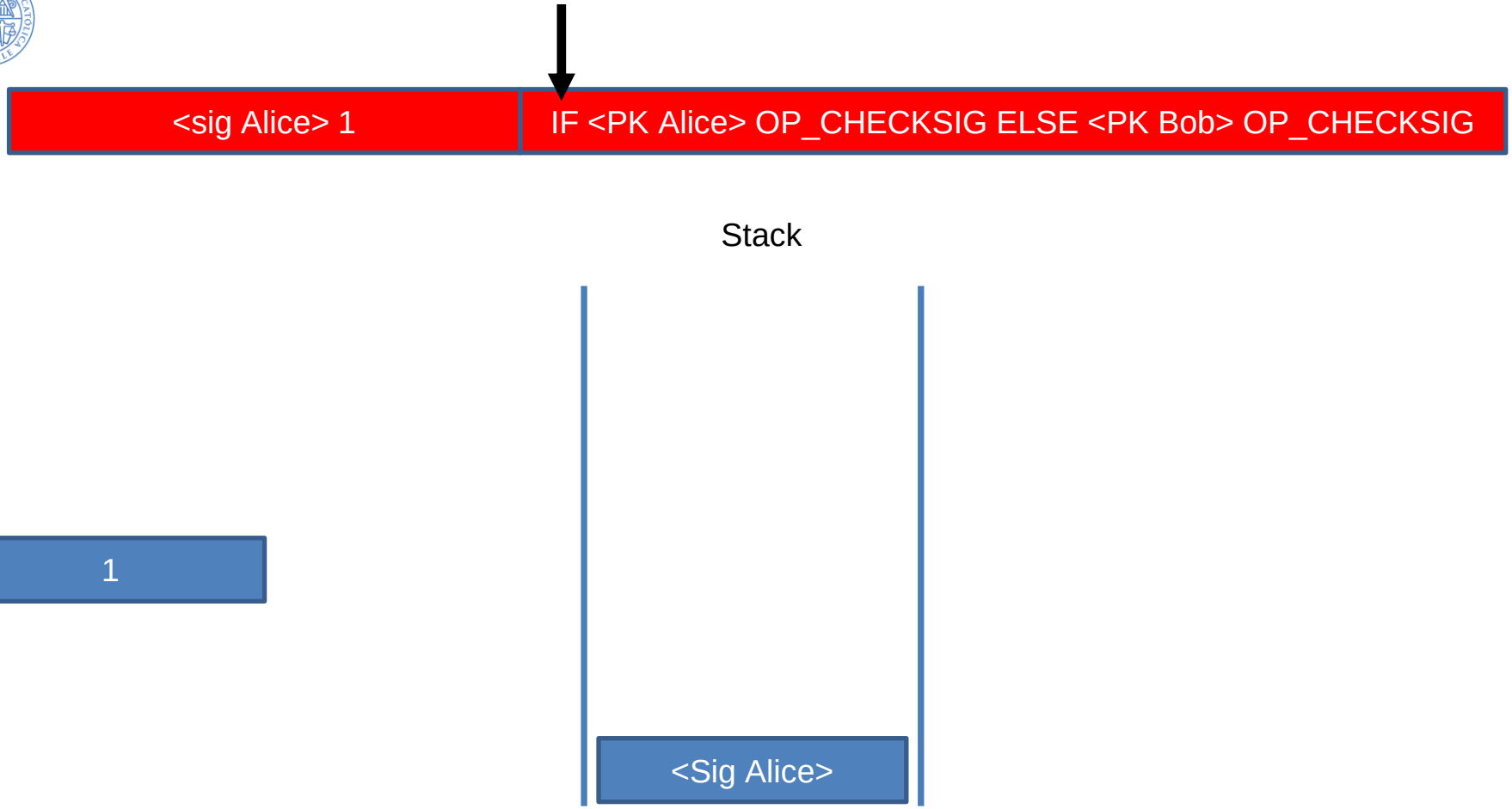


Multisig

Stack



Multisig


<sig Alice> 1
IF <PK Alice> OP_CHECKSIG ELSE <PK Bob> OP_CHECKSIG
Stack
1
<Sig Alice>

Multisig

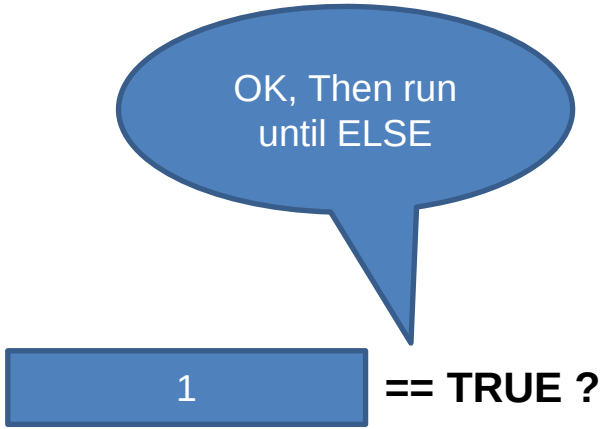
Stack

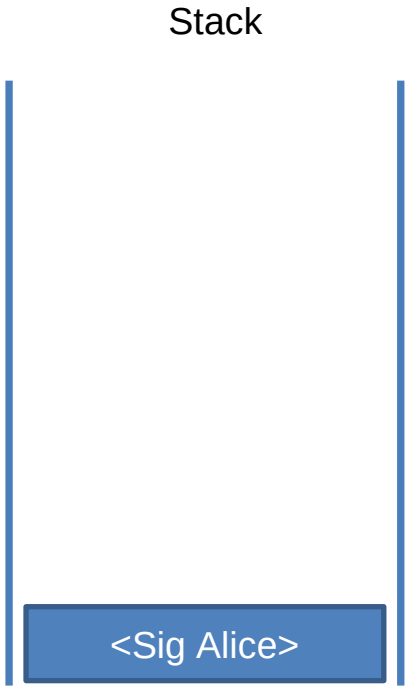
1 **== TRUE ?**

<Sig Alice>

Multisig


<sig Alice> 1
IF <PK Alice> OP_CHECKSIG ELSE <PK Bob> OP_CHECKSIG


OK, Then run
until ELSE
1
== TRUE ?


Stack
<Sig Alice>

Multisig

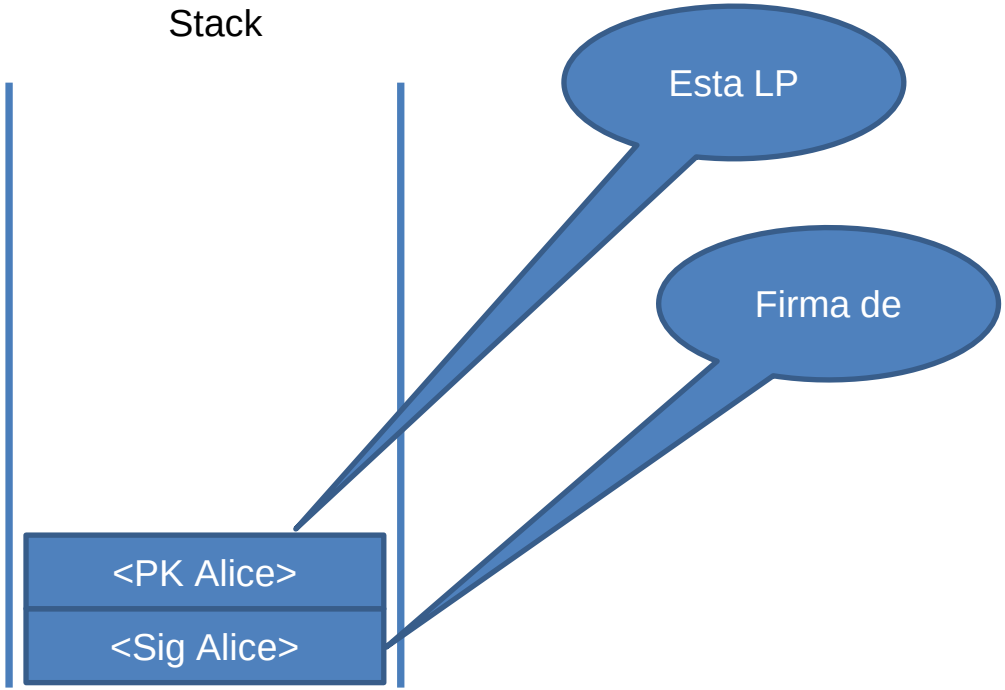
Stack

Multisig

Stack

Multisig

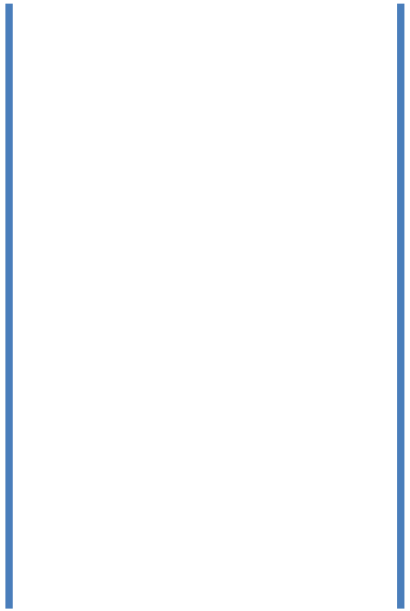

<sig Alice> 1
IF <PK Alice> OP_CHECKSIG ELSE <PK Bob> OP_CHECKSIG


Stack
Esta LP
Firma de
<PK Alice>
<Sig Alice>

Multisig

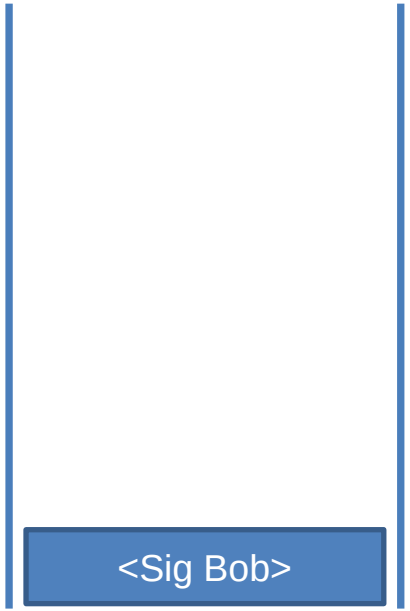
<sig Bob> 0	IF <PK Alice> OP_CHECKSIG ELSE <PK Bob> OP_CHECKSIG

Stack



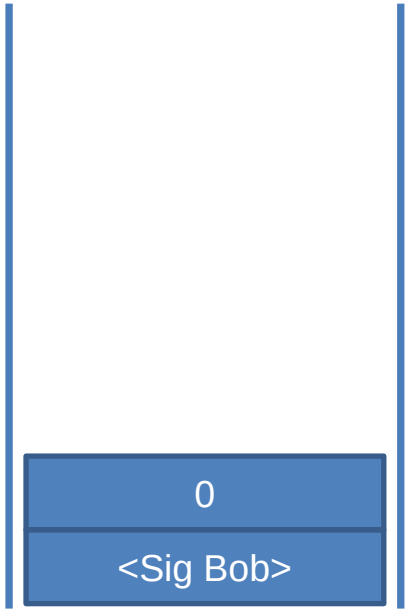
Multisig

Stack



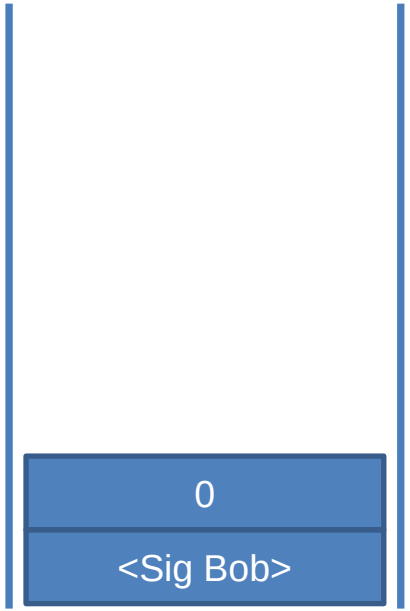
Multisig

Stack

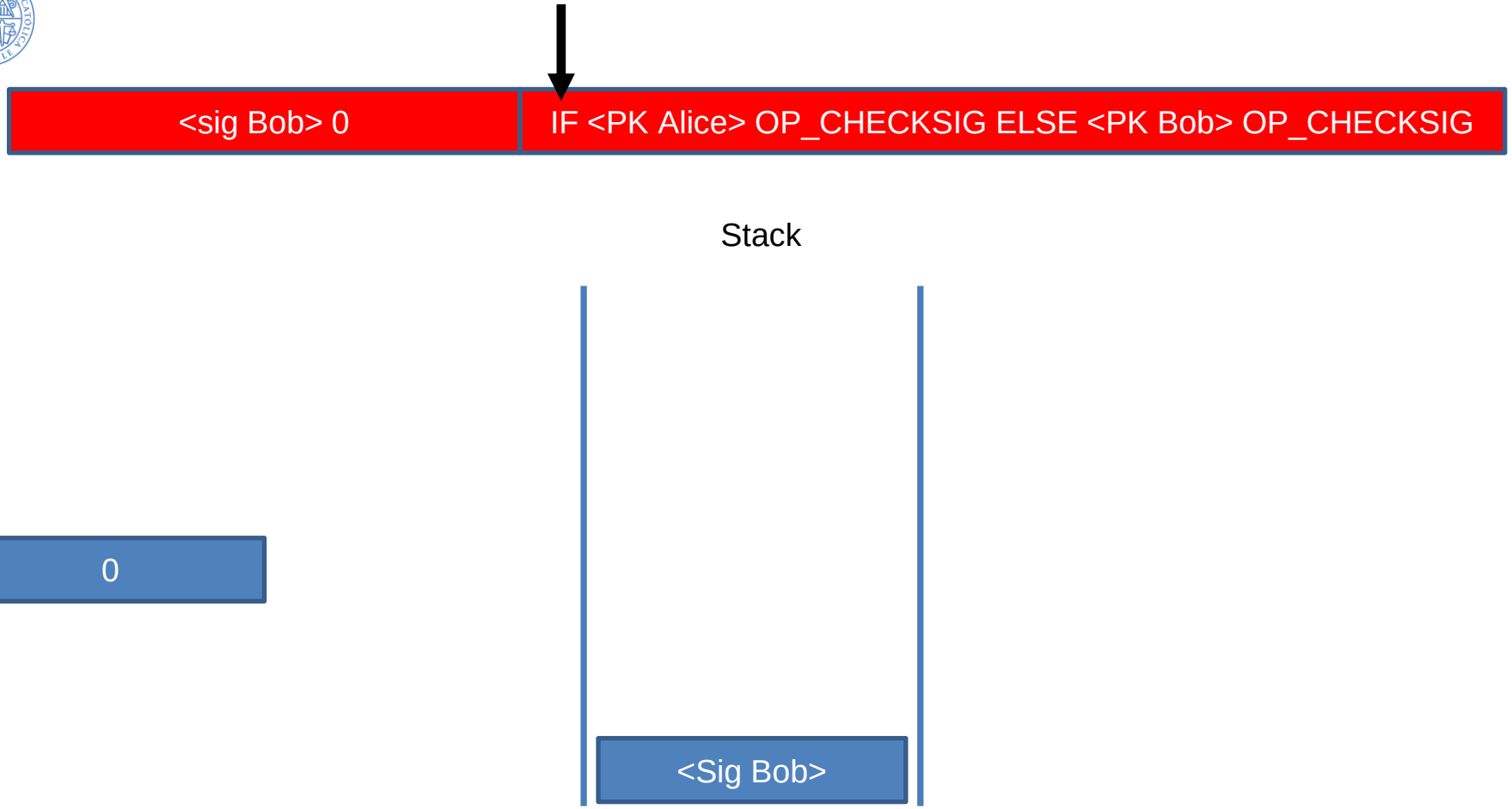


Multisig

Stack

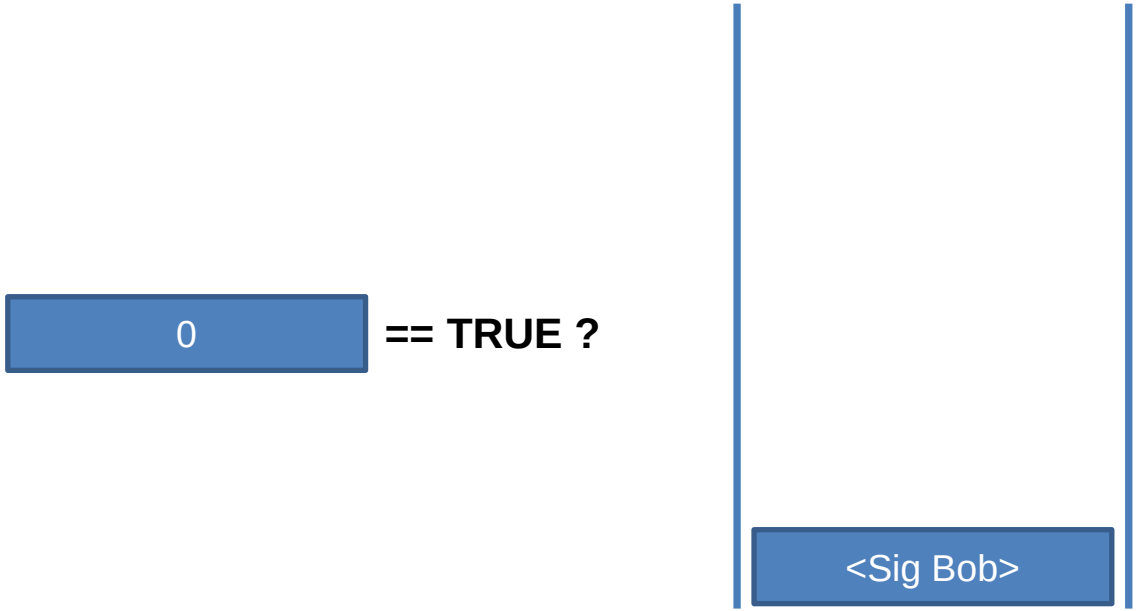


Multisig


<sig Bob> 0
IF <PK Alice> OP_CHECKSIG ELSE <PK Bob> OP_CHECKSIG
Stack
0
<Sig Bob>

Multisig

Stack



Multisig

NO, then run the ELSE clause

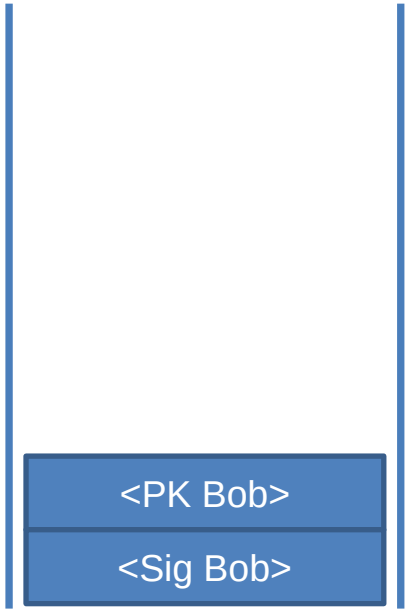
0 == TRUE ?

Stack

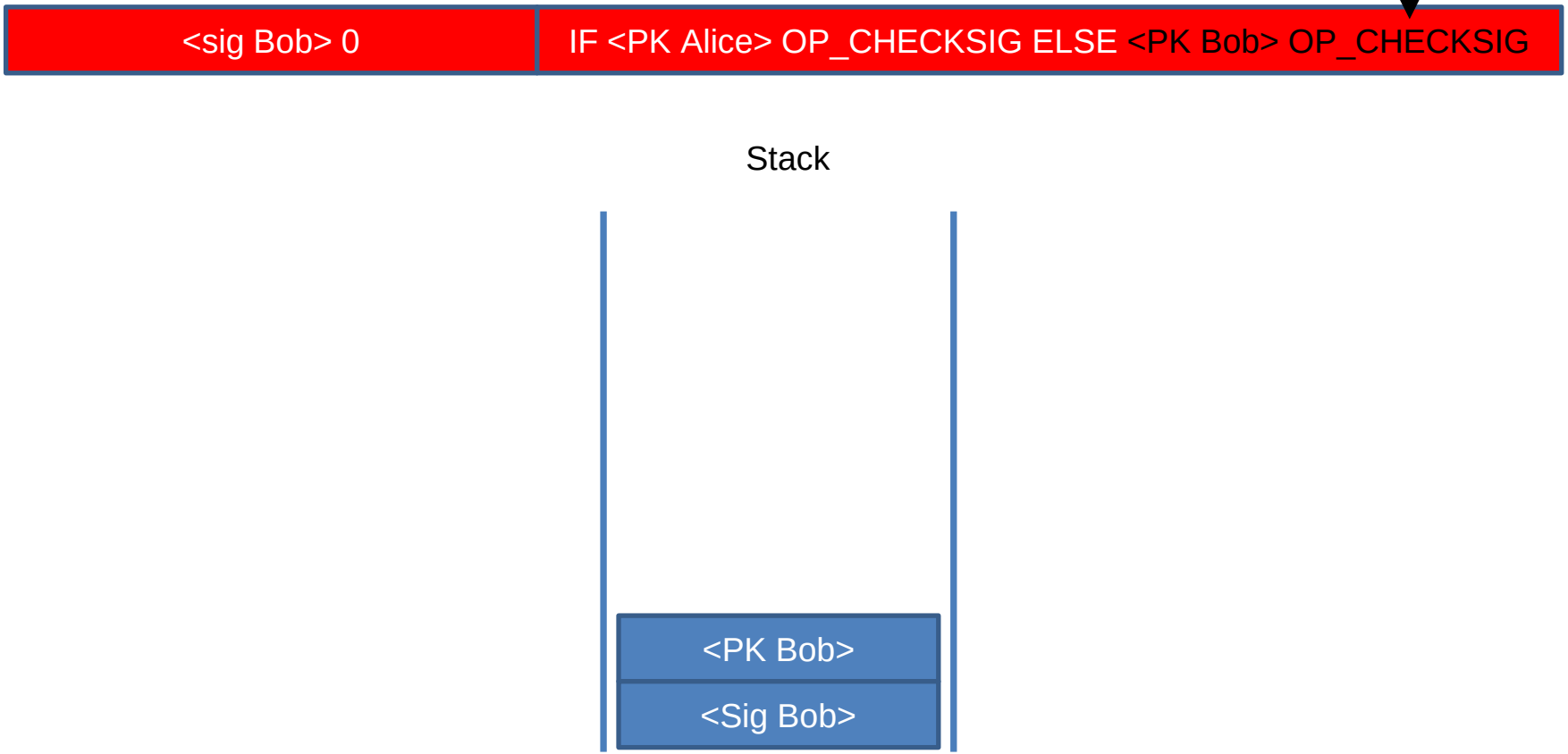
<Sig Bob>

Multisig

Stack

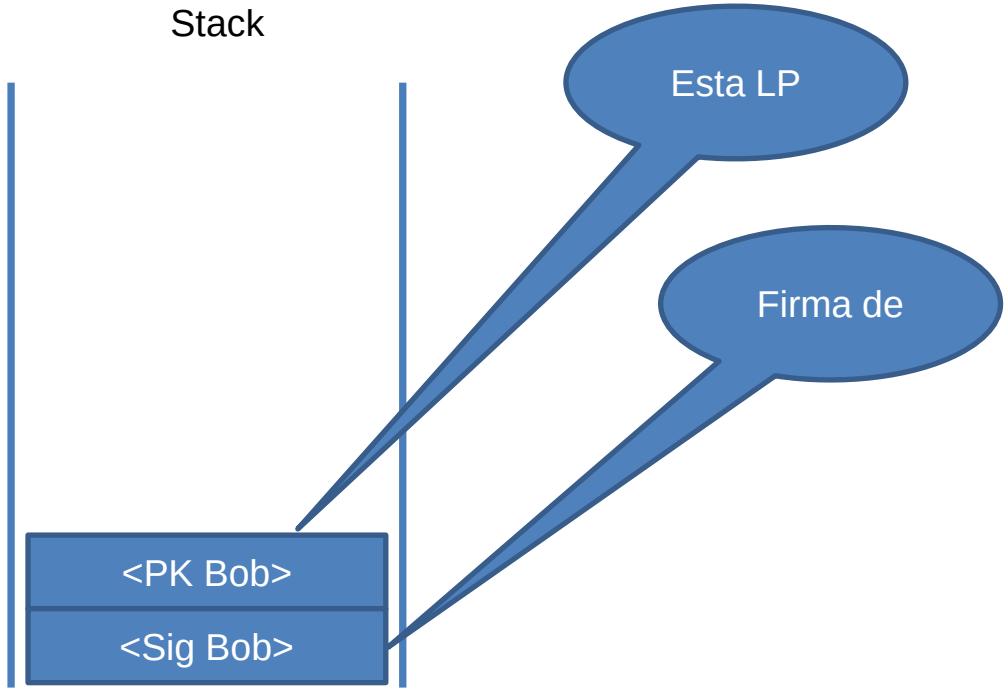


Multisig


<sig Bob> 0
IF <PK Alice> OP_CHECKSIG ELSE <PK Bob> OP_CHECKSIG
Stack
<PK Bob>
<Sig Bob>

Multisig

<sig Bob> 0 | IF <PK Alice> OP_CHECKSIG ELSE <PK Bob> OP_CHECKSIG



IF ELSE

Company with 3 owners, and a lawyer:

- Alice, Bob y Charlie son los socios
- Mohamed es el abogado
- Para gastar los fondos 2 de los 3 dueños deberías estar de acuerdo

Si no pueden ponerse de acuerdo entre 30 días, entra Mohamed:

- Para gastar los fondos se necesita la firma de Mohamed + uno de A,B,C

Si Alice, Bob y Charlie pierden sus llaves:

- Mohamed puede recuperar a los fondos después de 3 meses

IF ELSE

Company with 3 owners, and a lawyer

Locking script:

```
IF
  IF
    2
  ELSE
    <30 days> OP_CHECKSEQUENCEVERIFY OP_DROP
    <Lawyer's PK> OP_CHECKSIGVERIFY
    1
  ENDIF
  <PK Alice> <PK Bob> <PK Charlie> 3 OP_CHECKMULTISIG
ELSE
    <90 days> OP_CHECKSIGVERIFY OP_DROP
    <Lawyer's PK> OP_CHECKSIG
ENDIF
```

IF ELSE

3 caminos de ejecución:

```
IF
  IF
    2
  ELSE
    <30 days> OP_CHECKSEQUENCEVERIFY OP_DROP
    <Lawyer's PK> OP_CHECKSIGVERIFY
    1
  ENDIF
  <PK Alice> <PK Bob> <PK Charlie> 3 OP_CHECKMULTISIG
ELSE
    <90 days> OP_CHECKSIGVERIFY OP_DROP
    <Lawyer's PK> OP_CHECKSIG
ENDIF
```

IF ELSE

3 caminos de ejecución:

TRUE TRUE
(e.g. 0 <SIG A> <SIG C> TRUE TRUE)

```
IF
  IF
    2
  ELSE
    <30 days> OP_CHECKSEQUENCEVERIFY OP_DROP
    <Lawyer's PK> OP_CHECKSIGVERIFY
    1
  ENDIF
  <PK Alice> <PK Bob> <PK Charlie> 3 OP_CHECKMULTISIG
ELSE
    <90 days> OP_CHECKSIGVERIFY OP_DROP
    <Lawyer's PK> OP_CHECKSIG
ENDIF
```

IF ELSE

3 caminos de ejecución:

FALSE TRUE
(e.g. 0 <SIG Lawyer> <SIG C> FALSE TRUE)

```
IF
  IF
    2
  ELSE
    <30 days> OP_CHECKSEQUENCEVERIFY OP_DROP
    <Lawyer's PK> OP_CHECKSIGVERIFY
    1
  ENDIF
  <PK Alice> <PK Bob> <PK Charlie> 3 OP_CHECKMULTISIG
ELSE
    <90 days> OP_CHECKSIGVERIFY OP_DROP
    <Lawyer's PK> OP_CHECKSIG
ENDIF
```

IF ELSE

Unas preguntas:

- El abogado no puede robarse la plata en cualquier momento (FALSE)?
- Cuales caminos de ejecución se pueden usar 5, 35, 105 días después de la confirmación de esta transacción?
- Se pierden los fondos si el abogado pierde su llave?
- Como los socios hacen un reset cada 29 y 89 días para que el abogado no pueda robarse los fondos?

Canales de pago

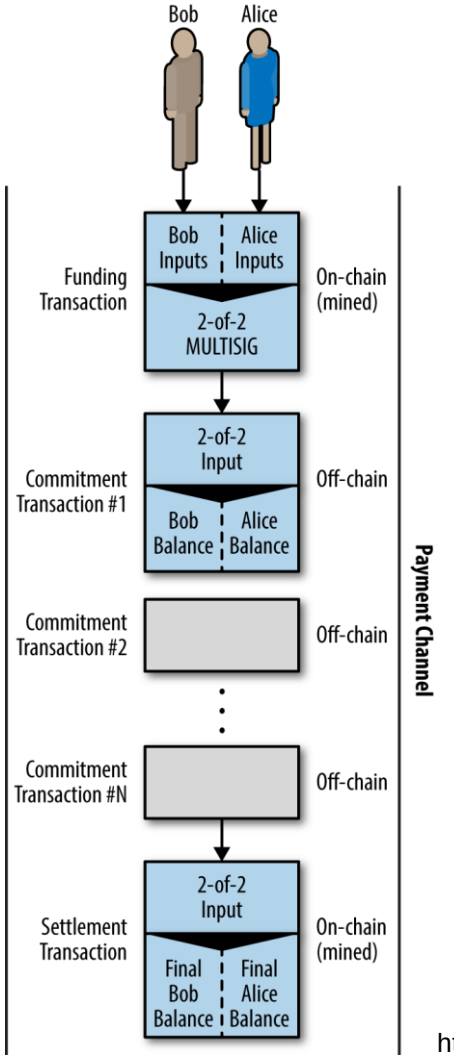
Un mecanismo para hacer transacciones off-chain:

- Transacciones instantáneas

Los dos partidos quieren ser de acuerdo sobre el balance de cada uno:

- Funding transaction (balance inicial)
- Una serie de commitment transactions (off-chain)
- Settlement transaction (balance final)

Canales de pago



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch12.asciidoc

Un canal simple

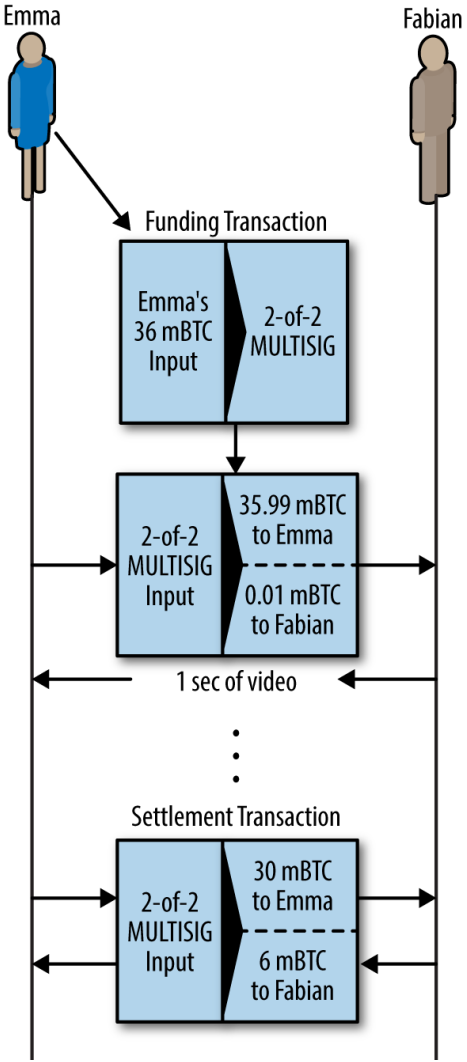
Streaming

Una aplicación de streaming:

- Fabian ofrece servicio de streaming y lo cobra por segundo
- Cada segundo cuesta 0.01mBTC = 0.00001 BTC
- Emma quiere ver X minutos de video
- Tienen software especial que permite esto a través canales de pago

Un canal simple

Streaming



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch12.asciidoc

Un canal simple

Streaming

- Emma manda commit#k con su firma (falta la firma de Fabian para gastar)
- Fabian devuelve 1 sec de video + commit#k con su firma (ya se puede gastar)

Puntos:

- Solo un commit se puede gastar (todos usan el mismo output como su input)
- Emma recibe la transacción n-1 firmada si la transmite: Fabian para stream!
- A él le conviene publicar el último

Un canal simple

Streaming

Problemas:

- Qué pasa si Fabian se retira antes de hacer primer commitment?
- Qué pasa si Emma transmite la commitment #1?

Estos problemas podemos solucionar usando timelocks

Un canal simple

Timelocked

Funding transaction:

- Emma construye la **funding** transaction, la firma, pero no la manda a Fabian
- Emma construye transacción de **refund** con timelock X (e.g. hoy + 30 días)
- Emma manda el refund a Fabian para firmar

- Un ves que tiene el refund transacition Emma transmite el funding
- Digamos que el funding tiene el timelock T + 4320 bloques (30 días)

- Si Fabian desaparece antes del Commit #1, Emma puede recuperar sus fondos con el refund

Idea: funding can actually be an IF/ELSE with a timelock refund to Emma

Un canal simple

Timelocked

Commitment transactions:

- (Funding: Emma paga 36mBTC a MULTISIG output de Emma y Fabian)
- Commit #1 paga 0.01mBTC a Fabian, y 35.99mBTC a Emma
- Commit #1 tiene el timelock de T + 4320 bloques

- Commit #2: F->0.02; E->35.98; T + 4319 bloques
- Commit #3: F->0.03; E->35.97; T + 4318 bloques
- Commit #4: F->0.04; E->35.96; T + 4317 bloques

Un canal simple

Timelocked

Commitment transactions:

- Commit #2: F->0.02; E->35.98; T + 4319 bloques
- Commit #3: F->0.03; E->35.97; T + 4318 bloques
- Commit #4: F->0.04; E->35.96; T + 4317 bloques

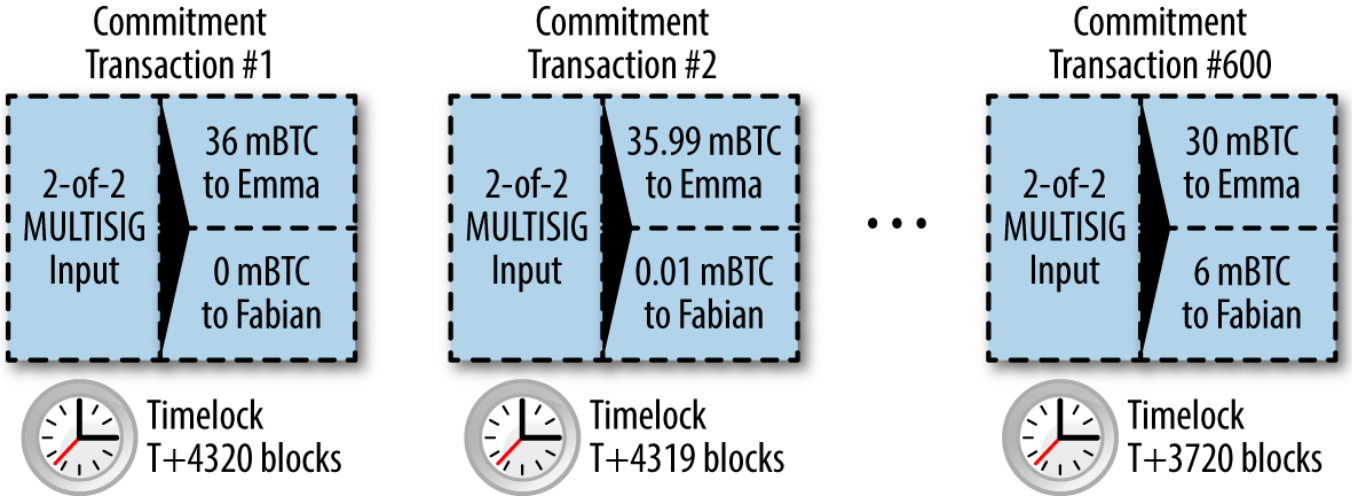
Emma puede transmitir Commit #2 y seguir viendo el video?

- Si no firma el Commit #3 NO, y Commit #3 se va a propagar antes de #2

Si todos se portan bien, firman el último commit sin el timelock y lo usan
Si no, cualquiera puede utilizar el último commit y cobrar su balance

Un canal simple

Timelocked



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch12.asciidoc

Timelocked channels

Lo bueno:

- Los dos partidos no necesitan confiar uno al otro
- No hay un intermediario
- Cada commit nuevo efectivamente invalida el previo

Lo malo:

- Hay un max timelock = un canal tiene vida finita (e.g. 4320 bloques)
- Hay un límite a numero de transacciones/commits (4320)

Bidirectional channels

Solución a los dos problemas

Puedo revocar el commit previo?

- Una vez válida, la transacción de Bitcoin nunca vence
- La única manera de qué no sea válida es si sus inputs se gastan
- Con timelocks hacemos precisamente esto, el último commit hace el "double spend" de las entradas del penúltimo commit

- Pero uno tiene que secuenciar todo bien
- Además, la plata va solo en una dirección (unidireccional)
- Y los otros problemas

Bidirectional channels

Solución a los dos problemas

La mejor solución:

- Canales bidireccionales
- Con llaves de revocación
- Que no expiran nunca (si los participantes no los cierran)

Parte con un funding a un multisig:

- Transacción donde Alice paga 5BTC, Bob paga 5BTC
- A un output que es un 2-2 MULTISIG de Alice + Bob
- (Se puede hacer el seguro para el refund como antes)

Bidirectional channels

Commit asimetrico

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimetrico

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimetrico

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimetrico

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Bidirectional channels

t asimetrico

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Bidirectional channels

Commit asimetrico

Commit #1 A

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Bidirectional channels

Commit asimetrico

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimetrico

Alice, fírmame esta wea porfa!

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimetrico

Alice, fírmame esta wea porfa!

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs: Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimetrico

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs: Alice

Bidirectional channels

Commit asimetrico

Commit #1 A

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Commit #1 B

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs: Alice

Bidirectional channels

Commit asimetrico

Commit #1 A

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob, Alice

Commit #1 B

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: <1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs: Alice, Bob

Bidirectional channels

Commit asimetrico revocable

Important: Slides 96 – 118 are not really how revocable commits work!

I am just using this simplified version in order to illustrate revocation keys!

- The original solution assumes that the secret key for a revocation thing is split!
- But the same logic applies!

Bidirectional channels

Commit asimetrico revocable

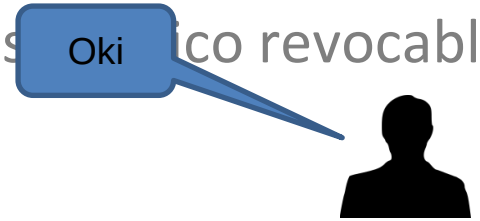
Revocation PK Alice#1
es mi secreto

Commit #1 Alice

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs:
```

Bidirectional channels

Commit as ico revocable



Commit #1 Alice

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs:
```

Bidirectional channels

Commit as[illegible]ico revocable

Bob, me puedes firmar esto porfa?

Oki

Commit #1 Alice

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob
```

Bidirectional channels

Commit asimetrico revocable

Revocation PK Alice#1
es mi secreto

Commit #1 Alice

Revocation PK Bob#1
es mi secreto

Commit #1 Bob

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob
```

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Bob#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG
Sigs:
```

Bidirectional channels

Commit asimetrico revocable

Revocation PK Alice#1
es mi secreto

Alice, necesito tu firma;
porfa ☺

Commit #1 Alice

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob
```

Commit #1 Bob

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Bob#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG
Sigs:
```

Bidirectional channels

Commit asimetrico revocable

Revocation PK Alice#1
es mi secreto

Commit #1 Alice

Revocation PK Bob#1
es mi secreto

Commit #1 Bob

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob
```

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Bob#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG
Sigs: Alice
```

Bidirectional channels

...netrico revocable

Si mando la tx a BTC, tengo que esperar 1000 bloques

Y a mi me pagan altiro; haha ☺

Commit #1 Alice

Commit #1 Bob

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob, Alice
```

Bidirectional channels

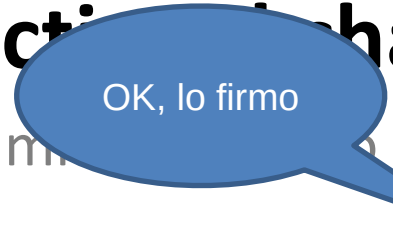
Commit asimétrico revocable

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs:
```

Bidirecti hannels

Comm revocable



Commit #2 Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG
Sigs:

Bidirecti hannels

Comm revocable

Pago 2 BTC a Bob

OK, lo firmo

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob
```

Bidirecti hannels

Comm revocable

Haha, me robo tu plata

Pucha ☹

Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG
Sigs: Bob, Alice

Para que sirve esto?

Bidirectional channels

o revocable

Pago 2 BTC a Bob

OK, mientras revocas el commit anterior te lo firmo

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs:
```

Bidirectional channels

o revocable

Pago 2 BTC a Bob

OK, mientras revocas el commit anterior te lo firmo

<Revocation PK Alice#1>

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs:
```

Bidirectional channels

Pago 2 BTC a Bob

Comm o revocable

Perfecto

<Revocation PK Alice#1>

Commit #2 Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG
Sigs: Bob

Bidirectional channels

Commit asimétrico revocable

Haha, me robo tu plata

<Revocation PK Alice#1>

Commit #1 Alice

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
          <999 blocks> CHECKSEQUENCEVERIFY
          <Revocation PK Alice#1>
        ELSE
          <1000 blocks>
          CHECKSEQUENCEVERIFY
          DROP
          <Alice PK>
          CHECKSIG
Sigs: Bob, Alice
```

Bidirectional channels

Commit asimétrico revocable

Haha, me robo tu plata

<Revocation PK Alice#1>

Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG
Sigs: Bob, Alice

Bidirectional channels

Commit asimétrico revocable

<Revocation PK Alice#1>

Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG
Sigs: Bob, Alice

Me llevo todo!

Bidirectional channels

Commit asimétrico revocable

Pago 2 BTC a Bob

<Revocation PK Alice#1>

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob
```

Bidirectional channels

Commit asimétrico revocable

Pago 2 BTC a Bob

<Revocation PK Alice#1>

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob
```

Commit #2 Bob

```
Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Bob#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG
Sigs:
```

Bidirectional channels

Commit asimétrico revocable

El revocation key#1
por favor!

<Revocation PK Alice#1>

Commit #2 Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG
Sigs: Bob

Commit #2 Bob

Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Bob#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG
Sigs:

Bidirectional channels

Commit asimétrico revocable

El revocation key#1 por favor!

<Revocation PK Alice#1>

<Revocation PK Bob#1>

Commit #2 Alice

Commit #2 Bob

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG
Sigs: Bob

Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Bob#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG
Sigs:

Bidirectional channels

Commit asimétrico revocable

Perfecto

<Revocation PK Alice#1>

<Revocation PK Bob#1>

Commit #2 Alice

Commit #2 Bob

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG
Sigs: Bob

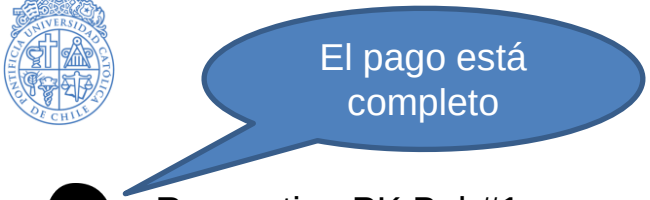
Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<999 blocks> CHECKSEQUENCEVERIFY
<Revocation PK Bob#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG
Sigs: Alice

Bidirectional channels

Commit asimétrico revocable



<Revocation PK Alice#1>

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG
Sigs: Bob
```

Commit #2 Bob

```
Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
            <999 blocks> CHECKSEQUENCEVERIFY
            <Revocation PK Bob#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG
Sigs: Alice
```

Bidirectional channels

Solución a los dos problemas

Preguntas

- **¿Alice puede publicar Commit #1? (¿Qué tiene Bob?)**

Idea:

- **Con la llave de revocación, a Bob/Alice ya no le conviene el estado anterior**

Detalle:

- **El partido a quien le están pagando tiene que recibir el revocation key antes de firmar**
- **A él siempre le conviene el último commit!**

Bidirectional channels

Commit asimétrico revocable

No tan bueno:

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2> EQUAL DROP
<Bob PK> CHECKSIG
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
ENDIF
CHECKSIG

Sigs: Bob

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2> EQUAL DROP
<Alice PK> CHECKSIG
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
ENDIF
CHECKSIG

Sigs: Alice

Bidirectional channels

Solución a los dos problemas

En realidad

- **Alice tiene mitad del secreto de revocación**
- **Bob tiene la otra mitad**

- **Lo intercambian cuando hacen el revoke**

- **Esto asegura que un partido si se retira, el otro puede gastar el commit anterior con el revoke1 + revoke2**

Bidirectional channels

Commit asimétrico revocable

Alice tiene mitad de la llaves secretas A y B

Commit Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK A> CHECKSIG
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Commit Bob

nput: 2-2 MULTISIG

Outpu <3BTC>
Lock 0: < PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<Revocation PK B> CHECKSIG
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimétrico revocable

Alice tiene mitad de la llaves secretas A y B

Commit Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <Revocation PK A> CHECKSIG
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG

Sigs: Bob
```

Commit Bob

```
Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
            <Revocation PK B> CHECKSIG
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG

Sigs:
```

Bidirectional channels

Commit asimétrico revocable

Alice tiene mitad de la llaves secretas A y B

Commit Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK A> CHECKSIG
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Bob tiene la otra mitad

Commit Bob

Input: 2-

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<Revocation PK B> CHECKSIG
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimétrico revocable

Revoke es la mitad de la llave secreta!

Commit Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <Revocation PK A> CHECKSIG
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG

Sigs: Bob
```

Commit Bob

```
Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
            <Revocation PK B> CHECKSIG
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG

Sigs:
```

Bidirectional channels

Solución a los dos problemas

Importante:

- **La revocación no ocurre automáticamente**
- **Los dos partidos deberían ser atento a las transacciones en la red**
- **Básicamente, necesitan correr un nodo (puede ser SPV)**

Problema con todo esto:

- **Como combinar dos canales de pago?**
- **HTLC (Hashed Timelock Contract; BIP 112)**

Lightning Network!!!

HTLC

Hashed TimeLock Contract (HTLC):

- **Alice y Bob tienen su canal de pago**
- **Alice paga 1 BTC a Bob, si Bob le muestra un secreto en 1 día**
- **Si no, Alice puede recuperar su 1 BTC**

"HTLC Output"

HTCL

Canal A<->B, cada uno tiene 5 BTC

A manda 1 BTC a B con HTCL de 1 dia

secret s.t. H(secret) = secretHash

Commit Alice:

Input: 2-2 MULTISIG

Output 0: <5BTC> → <PK Bob>
Output 1: <4BTC> → <PK Alice> (+1000 block delay)

Output 2: <1BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+1day> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

Commit Bob:

Input: 2-2 MULTISIG

Output 0: <4BTC> → <PK Alice>
Output 1: <5BTC> → <PK Bob> (+1000 block delay)

Output 2: <1BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+1day> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Alice

HTCL

Caso #1: Bob publica la transacción

Puede cobrar Output #2 con el secret (antes de 1 día)

Commit Alice:

Input: 2-2 MULTISIG

Output 0: <5BTC> → <PK Bob>
Output 1: <4BTC> → <PK Alice> (+1000 block delay)

Output 2: <1BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+1day> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

secret

Commit Bob:

Input: 2-2 MULTISIG

Output 0: <4BTC> → <PK Alice>
Output 1: <5BTC> → <PK Bob> (+1000 block delay)

Output 2: <1BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+1day> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Alice

HTCL

Caso #2: Alice publica la transacción

Puede cobrar Output #2 despues de un día

Commit Alice:

Input: 2-2 MULTISIG

Output 0: <5BTC> → <PK Bob>
Output 1: <4BTC> → <PK Alice> (+1000 block delay)

Output 2: <1BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+1day> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

secret

Commit Bob:

Input: 2-2 MULTISIG

Output 0: <4BTC> → <PK Alice>
Output 1: <5BTC> → <PK Bob> (+1000 block delay)

Output 2: <1BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+1day> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Alice

HTCL

Canal A<->B, A tiene 10 BTC, B 8BTC

Canal B <-> C, B tiene 5BTC, C 3BTC

HTCL

Canal A<->B, A tiene 10 BTC, B 8BTC

Canal B <-> C, B tiene 5BTC, C 3BTC

Quiero mandar 1BTC
a Charlie

HTCL

Canal A<->B, A tiene 10 BTC, B 8BTC

Canal B <-> C, B tiene 5BTC, C 3BTC

secret

Quiero mandar 1BTC
a Charlie

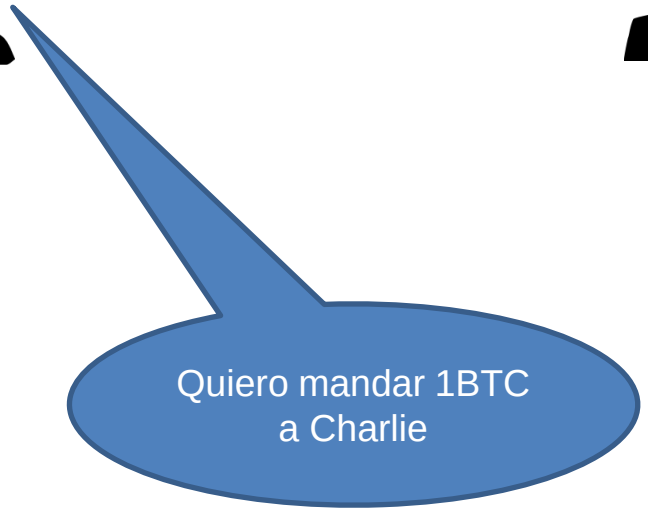
HTCL

Canal A<->B, A tiene 10 BTC, B 8BTC

Canal B <-> C, B tiene 5BTC, C 3BTC

secret

secretHash = h(secret)



HTCL

Canal A<->B, A tiene 10 BTC, B 8BTC

Canal B <-> C, B tiene 5BTC, C 3BTC

secret

secretHash = h(secret)

Quiero mandar 1BTC a Charlie

Pagale a Bob 1BTC a HTLC output si te muestra el secret

HTCL

Canal A<->B, A tiene 10 BTC, B 8BTC

Canal B <-> C, B tiene 5BTC, C 3BTC

Commit Alice (A-B)

secret

secretHash = h(secret)

Input: 2-2 MULTISIG (A,B)

Output 0: <8BTC> → <PK Bob>
Output 1: <8.9 BTC> → <PK Alice> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+5h> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

HTCL

Charlie te va a decir el secret si le pagas 1 BTC

Canal A<-> [illegible]

Canal B <-> C, B tiene 5BTC, C 3BTC

secret

Commit Alice (A-B)

secretHash = h(secret)

Input: 2-2 MULTISIG (A,B)

Output 0: <8BTC> → <PK Bob>
Output 1: <8.9 BTC> → <PK Alice> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+5h> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

HTCL

Commit Alice (A-B)

Input: 2-2 MULTISIG (A,B)

Output 0: <8BTC> → <PK Bob>
Output 1: <8.9 BTC> → <PK Alice> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+5h> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

secretHash = h(secret)

HTCL

Charlie te va a decir el secret si le pagas 1 BTC

Canal A<-> [illegible]

Canal B <-> C, B tiene 5BTC, C 3BTC

secret

Commit Alice (A-B)

Input: 2-2 MULTISIG (A,B)

Output 0: <8BTC> → <PK Bob>
Output 1: <8.9 BTC> → <PK Alice> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+5h> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

Commit Bob (B-C)

Input: 2-2 MULTISIG (C,B)

Output 0: <3BTC> → <PK Charlie>
Output 1: <3.9 BTC> → <PK Bob> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Charlie>
ELSE
<HTLC timeout = T+4h> CHECKLOCKTIMEVERIFY
<PK Bob>
ENDIF
CHECKSIG

Sigs: Charlie

HTCL

Canal A<->B, A tiene 10 BTC, B 8BTC

Canal B <-> C, B tiene 5BTC, C 3BTC

Commit Alice (A-B)

Commit Bob (B-C)

secret

Input: 2-2 MULTISIG (A,B)

Output 0: <8BTC> → <PK Bob>
Output 1: <8.9 BTC> → <PK Alice> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+5h> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

Input: 2-2 MULTISIG (C,B)

Output 0: <3BTC> → <PK Charlie>
Output 1: <3.9 BTC> → <PK Bob> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Charlie>
ELSE
<HTLC timeout = T+4h> CHECKLOCKTIMEVERIFY
<PK Bob>
ENDIF
CHECKSIG

Sigs: Charlie

HTCL

Charlie, Secret es "secret"

Canal A<->B, A tiene 10 BTC, B 8BTC

Canal B <-> C, B tiene 5BTC, C [illegible]BTC

Commit Alice (A-B)

Commit Bob (B-C)

secret

Input: 2-2 MULTISIG (A,B)

Output 0: <8BTC> → <PK Bob>
Output 1: <8.9 BTC> → <PK Alice> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+5h> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

Input: 2-2 MULTISIG (C,B)

Output 0: <3BTC> → <PK Charlie>
Output 1: <3.9 BTC> → <PK Bob> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Charlie>
ELSE
<HTLC timeout = T+4h> CHECKLOCKTIMEVERIFY
<PK Bob>
ENDIF
CHECKSIG

Sigs: Charlie

HTCL

Alice, Secret es "secret"

Canal A<->B, A tiene 10 BTC, B tiene 8BTC

Charlie, Secret es "secret"

Canal B <-> C, B tiene 5BTC, C tiene 3BTC

secret

Commit Alice (A-B)

Input: 2-2 MULTISIG (A,B)

Output 0: <8BTC> → <PK Bob>
Output 1: <8.9 BTC> → <PK Alice> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Bob>
ELSE
<HTLC timeout = T+5h> CHECKLOCKTIMEVERIFY
<PK Alice>
ENDIF
CHECKSIG

Sigs: Bob

Commit Bob (B-C)

Input: 2-2 MULTISIG (C,B)

Output 0: <3BTC> → <PK Charlie>
Output 1: <3.9 BTC> → <PK Bob> (+1000 block delay)

Output 2: <1.1 BTC>
HASH160 <secretHash> EQUAL
IF
<PK Charlie>
ELSE
<HTLC timeout = T+4h> CHECKLOCKTIMEVERIFY
<PK Bob>
ENDIF
CHECKSIG

Sigs: Charlie

HTLC

Nos falta una cosa:

- **Implementar HTLC con revocation keys**
- **Se aplica la misma lógica de antes**
- **Pero las transacciones son un poco distintas**

HTLC

HTLC se puede gastar si:

- **El destinatario conoce el secreto – fondos van al destinatario**
- **El HTLC hico el time-out (Locktime) – fondos van al sender**
- **El commit fue revocado – se castiga al tramposo**

Canal A<->B, cada uno tiene 5 BTC

A manda 1 BTC a B con HTCL

HTCL

Commit sender:

Input: 2-2 MULTISIG

```
Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG
Output 1: <4BTC>
Lock 1: IF
            <Revocation PK Alice>
          ELSE
            <1000 blocks> CHECKSEQUENCEVERIFY  DROP
            <Alice PK> CHECKSIG

Output 2: <1BTC>
              H160 <secretHash> EQUAL SWAP
              <Revocation PK Alice> EQUAL ADD (that is, OR)
              IF
                 <PK Bob>
              ELSE
                  <HTLC timeout = T+1day> CHECKLOCKTIMEVERIFY
                  <1000 blocks> CHECKSEQUENCEVERIFY DROP
                  <PK Alice>
              ENDIF
              CHECKSIG

Sigs: Bob
```

Commit receiver:

Input: 2-2 MULTISIG

```
Output 0: <4BTC>
Lock 0: <Alice PK> CHECKSIG
Output 1: <5BTC>
Lock 1: IF
            <Revocation PK Bob>
          ELSE
            <1000 blocks> CHECKSEQUENCEVERIFY  DROP
            <Bob PK> CHECKSIG

Output 2: <1BTC>
              H160 <secretHash> EQUAL
              IF
                  <1000 blocks> CHECKSEQUENCEVERIFY DROP
                  <PK Bob>
              ELSE
                  <Revocation PK Bob> EQUAL
                  NOTIF
                  <HTLC timeout = T+1day> CHECKLOCKTIMEVERIFY
                  ENDIF
                  <PK Alice>
              ENDIF
              CHECKSIG

Sigs: Alice
```

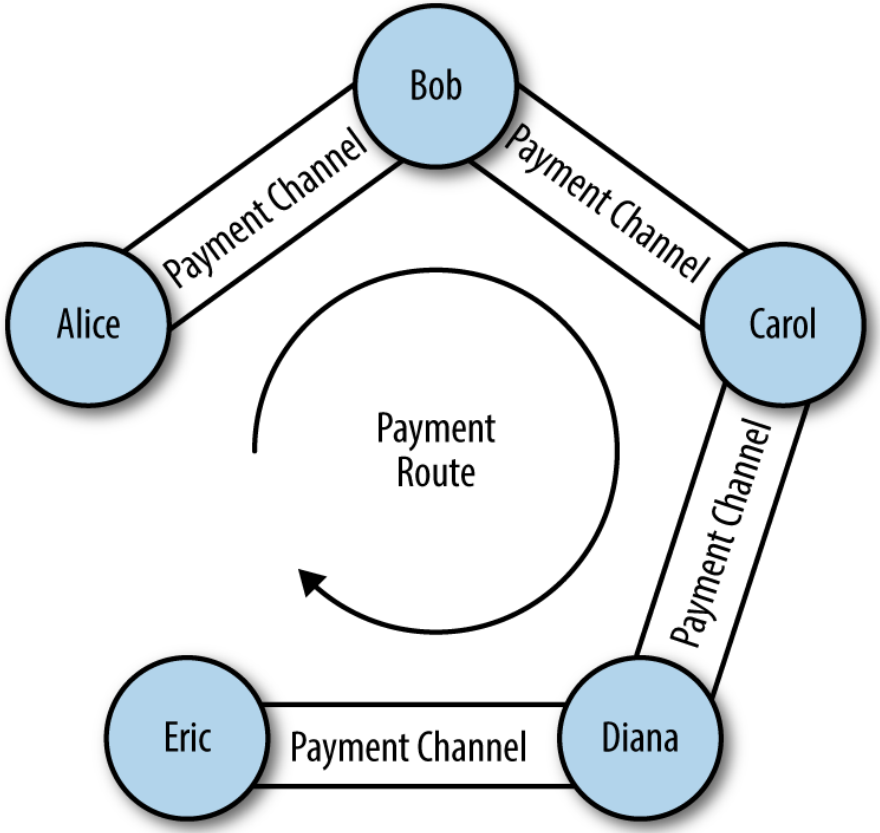
Lightning network

Lightning network:

- Nodos corriendo varios canales de pago
- A paga a B tal que encuentra una ruta con suficiente capacidad en la red
- Y construye un commit en su primer canal de pago, con info para proceder

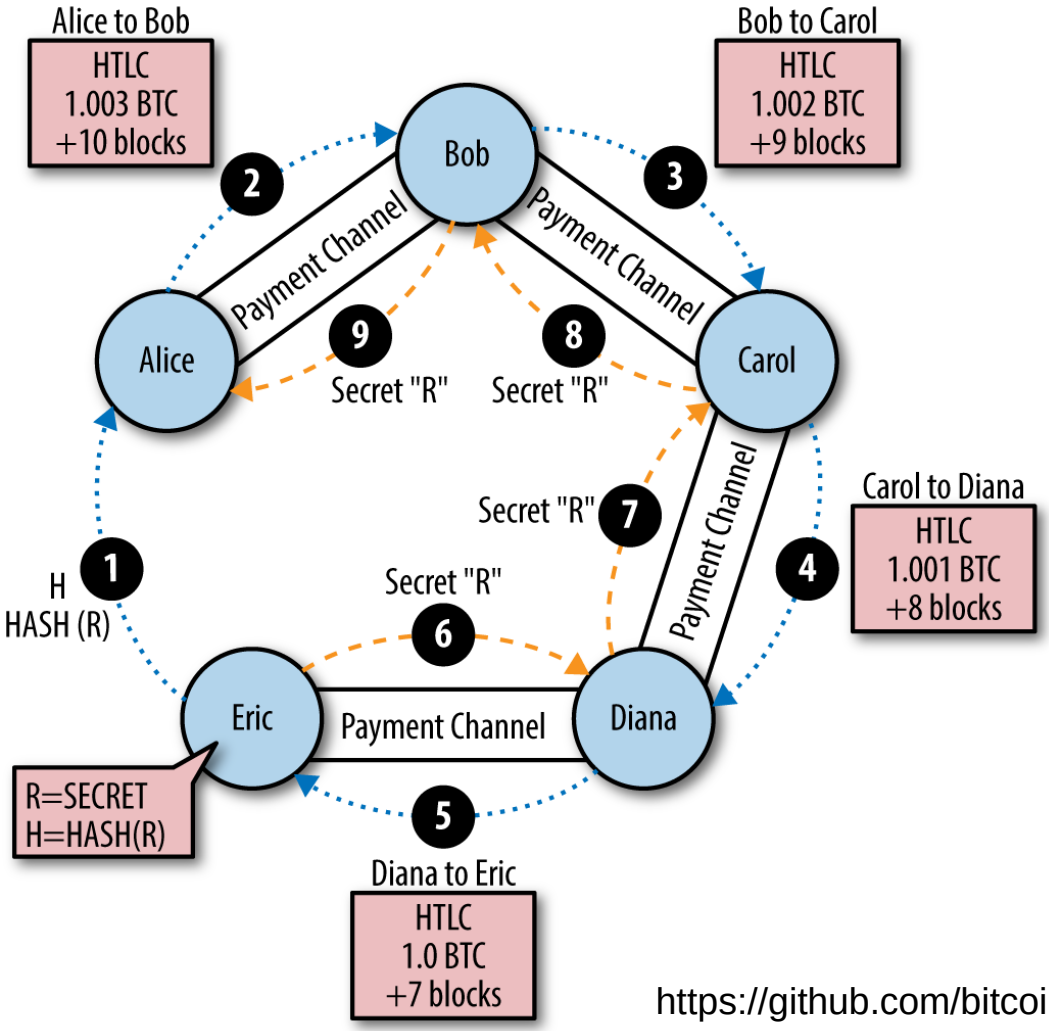
Lightning network

Lightning network



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch12.asciidoc

Lightning network



Alice paga 1 BTC a Eric

Eric genera el secreto R

Todo el resto: HTLC

https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch12.asciidoc

Lightning network

Lightning network benefits:

- Velocidad
- Privacidad
- Fungibilidad
- Granularidad (puedo pagar hasta 1 Satoshi)
- Capacidad
- Todavía trustless (como Bitcoin)

Lightning network

Resources for Lightning:

- https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch12.asciidoc
- https://github.com/ElementsProject/lightning/blob/master/doc/deployable-lightning.pdf
- https://github.com/bitcoin/bips/blob/master/bip-0112.mediawiki#Hash_TimeLocked_Contracts
- http://lightning.network/lightning-network-paper.pdf
- https://github.com/ChristopherA/Learning-Bitcoin-from-the-CommandLine/blob/master/11_3_Empowering_Bitcoin_with_Scripts.md

Bidirectional channels

Commit asimetrico revocable

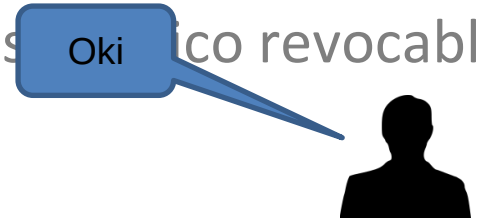
Revocation PK Alice#1
es mi secreto

Commit #1 Alice

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG

Sigs:
```

Bidirectional channels

Commit as[illegible]ico revocable



Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

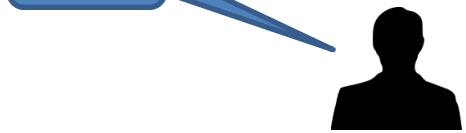
Output 1: <5BTC>
Lock 1: IF
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit as[illegible]ico revocable

Bob, me puedes firmar esto porfa?



Oki

Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Bidirectional channels

Commit asimetrico revocable

Revocation PK Alice#1
es mi secreto

Commit #1 Alice

Revocation PK Bob#1
es mi secreto

Commit #1 Bob

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<Revocation PK Bob#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimetrico revocable

Revocation PK Alice#1
es mi secreto

Alice, necesito tu firma;
porfa ☺

Commit #1 Alice

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG

Sigs: Bob
```

Commit #1 Bob

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <Revocation PK Bob#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG

Sigs:
```

Bidirectional channels

Commit asimetrico revocable

Revocation PK Alice#1
es mi secreto

Commit #1 Alice

Revocation PK Bob#1
es mi secreto

Commit #1 Bob

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<Revocation PK Bob#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs: Alice

Bidirectional channels

netrico revocable

Si mando la tx a BTC, tengo que esperar 1000 bloques

Y a mi me pagan altiro; haha ☺

Commit #1 Alice

Commit #1 Bob

```
Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
            <Revocation PK Alice#1>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG

Sigs: Bob, Alice
```

Bidirectional channels

Commit asimétrico revocable

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG

Sigs:
```

Bidirecti[illegible]hannels

Comm[illegible] revocable

Commit #2 Alice

Input: 2-2 MULTISIG

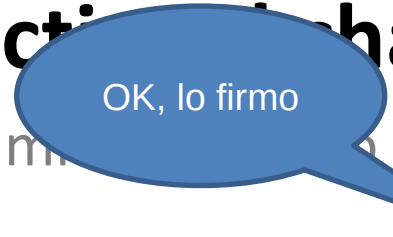
Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Bidirecti hannels

Comm revocable



Commit #2 Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Bidirecti hannels

Comm revocable

Haha, me robo tu plata

Pucha ☹

Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob, Alice

Para que sirve esto?

Bi...l channels

...o revocable

Pago 2 BTC a Bob

OK, mientras revocas el commit anterior te lo firmo

Commit #2 Alice

Input: 2-2 MULTISIG

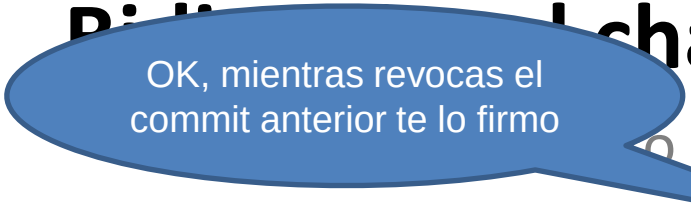
Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Bidirectional channels

o revocable



<Revocation PK Alice#1>

Commit #2 Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs:

Bidirectional channels

Comm o revocable

<Revocation PK Alice#1>

Commit #2 Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Bidirectional channels

Commit asimétrico revocable

Haha, me robo tu plata

<Revocation PK Alice#1>

Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob, Alice

Bidirectional channels

Commit asimétrico revocable

Haha, me robo tu plata

<Revocation PK Alice#1>

Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
<Revocation PK Alice#1>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob, Alice

Bidirectional channels

Commit asimétrico revocable

<Revocation PK Alice#1>

Commit #1 Alice

Input: 2-2 MULTISIG

Output 0: <5BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <5BTC>
Lock 1: IF
  <Revocation PK Alice#1>
ELSE
  <1000 blocks>
  CHECKSEQUENCEVERIFY
  DROP
  <Alice PK>
  CHECKSIG

Sigs: Bob, Alice

Me llevo todo!

Bidirectional channels

Commit asimétrico revocable

Pago 2 BTC a Bob

<Revocation PK Alice#1>

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG

Sigs: Bob
```

Bidirectional channels

Commit asimétrico revocable

Pago 2 BTC a Bob

<Revocation PK Alice#1>

Commit #2 Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Commit #2 Bob

Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<Revocation PK Bob#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimétrico revocable

El revocation key#1 por favor!

<Revocation PK Alice#1>

Commit #2 Alice

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Commit #2 Bob

Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<Revocation PK Bob#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimétrico revocable

El revocation key#1 por favor!

<Revocation PK Alice#1>

<Revocation PK Bob#1>

Commit #2 Alice

Commit #2 Bob

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<Revocation PK Bob#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs:

Bidirectional channels

Commit asimétrico revocable

Perfecto

<Revocation PK Alice#1>

<Revocation PK Bob#1>

Commit #2 Alice

Commit #2 Bob

Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
<Revocation PK Alice#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Alice PK>
CHECKSIG

Sigs: Bob

Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
<Revocation PK Bob#2>
ELSE
<1000 blocks>
CHECKSEQUENCEVERIFY
DROP
<Bob PK>
CHECKSIG

Sigs: Alice

Bidirectional channels

Commit asimétrico revocable

El pago está completo

<Revocation PK Bob#1>

<Revocation PK Alice#1>

Commit #2 Alice

```
Input: 2-2 MULTISIG

Output 0: <7BTC>
Lock 0: <Bob PK> CHECKSIG

Output 1: <3BTC>
Lock 1: IF
            <Revocation PK Alice#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Alice PK>
            CHECKSIG

Sigs: Bob
```

Commit #2 Bob

```
Input: 2-2 MULTISIG

Output 0: <3BTC>
Lock 0: <Alice PK> CHECKSIG

Output 1: <7BTC>
Lock 1: IF
            <Revocation PK Bob#2>
        ELSE
            <1000 blocks>
            CHECKSEQUENCEVERIFY
            DROP
            <Bob PK>
            CHECKSIG

Sigs: Alice
```