

Transactions, UTXO and the Bitcoin Application Layer

Introduction to Blockchain Science and Engineering

Aggelos Kiayias

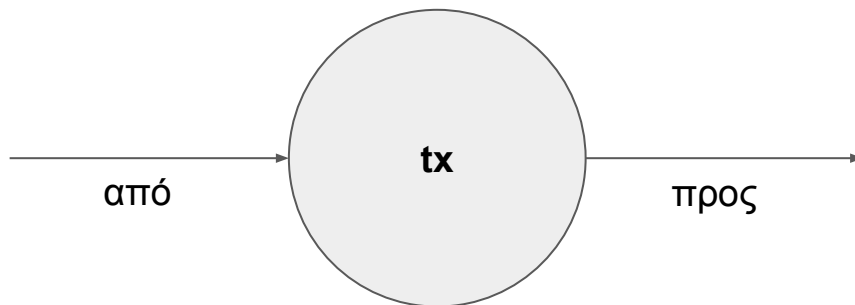
Dionysis Zindros, Christos Nasikas

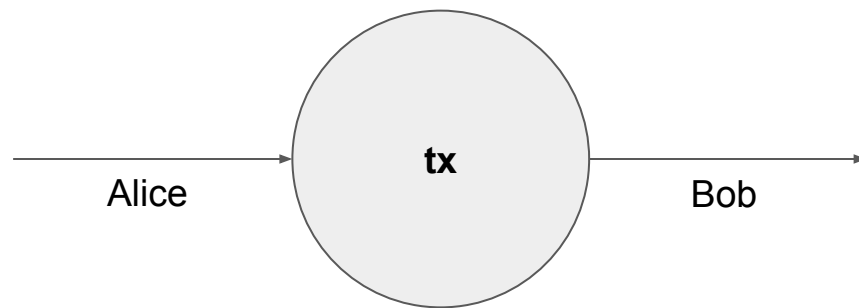
Στόχος του μαθήματος

- Συναλλαγές, ρέστα
- Γράφος του bitcoin, ακμές, κόμβοι, αξίες, ιδιοκτήτες, utxo, coinbase
- Εξόρυξη, consensus, blockchain, genesis στο bitcoin
- Bitcoin script
- p2pk, p2pkh, multisig

Συναλλαγές

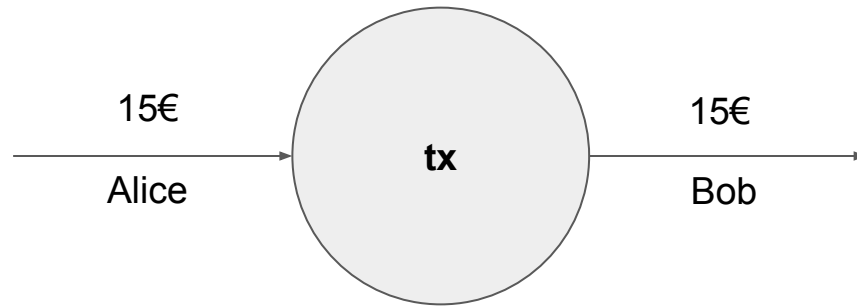
- Η βασική δομή του bitcoin είναι η **συναλλαγή** (transaction - tx)
- Μία συναλλαγή μεταφέρει χρήματα από έναν κάτοχο σε έναν άλλον

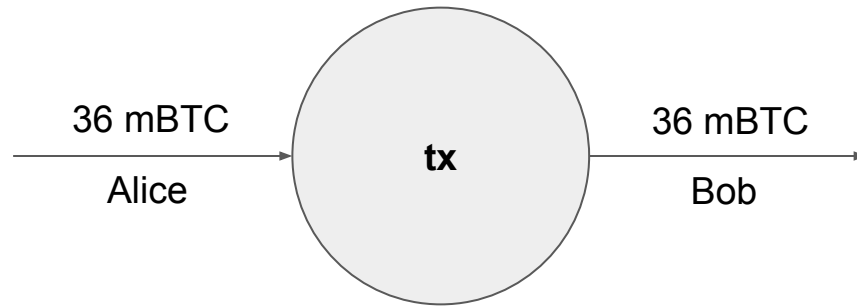




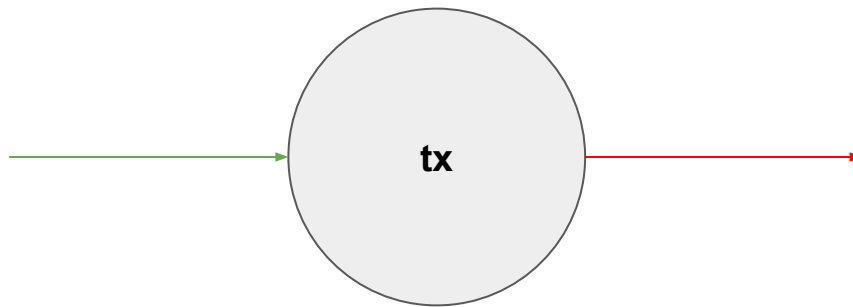
Ακμές συναλλαγών

- Μία συναλλαγή αναπαρίσταται από έναν **κόμβο**
- Έχει **εισερχόμενες** και **εξερχόμενες ακμές**
- Η εισερχόμενη ακμή αντιπροσωπεύει **ποιος πληρώνει**
- Η εξερχόμενη ακμή αντιπροσωπεύει **ποιος πληρώνεται**
- Οι κόμβοι **δεν** αντιπροσωπεύουν ιδιοκτήτες, αλλά συναλλαγές
- **Οι ακμές έχουν ιδιοκτήτες**
- Κάθε ακμή έχει ένα **βάρος** που αποτελεί την οικονομική αξία της





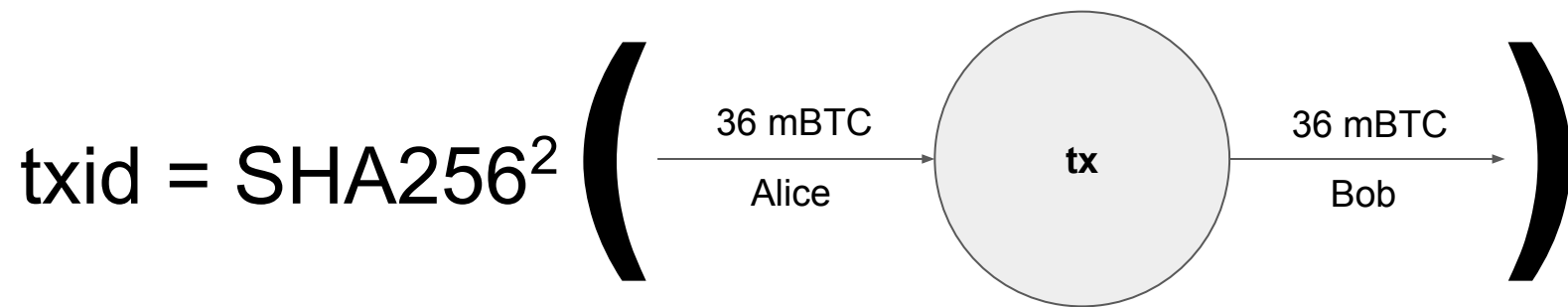
είσοδος / input

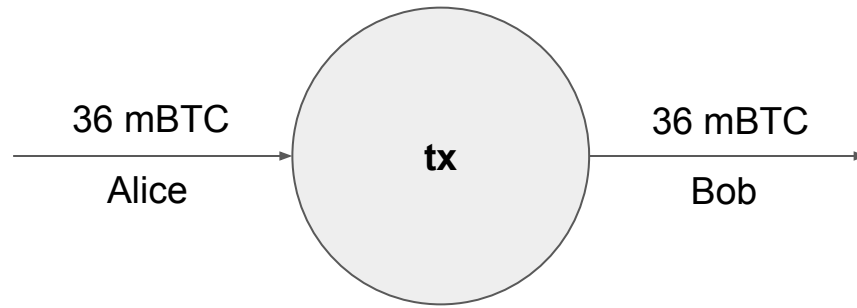


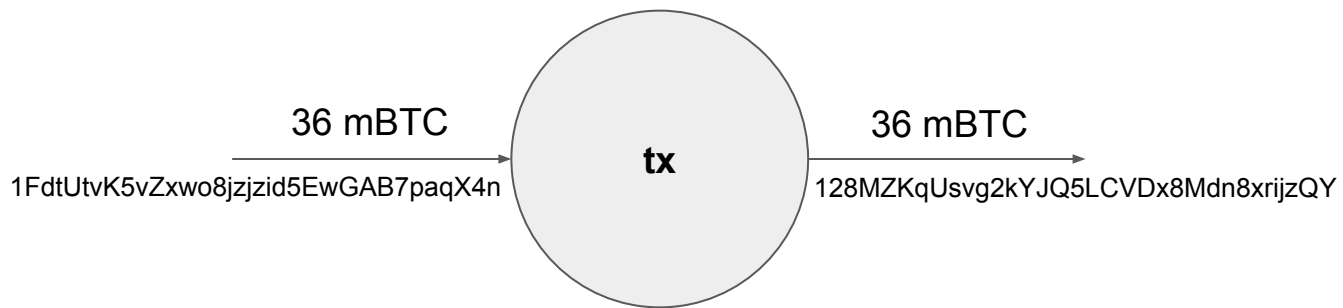
έξοδος / output

Δημόσιες συναλλαγές

- Όλες οι συναλλαγές δημοσιεύονται!
- Καθένας μπορεί να δει όλες τις συναλλαγές
- **Ανωνυμία** επιτυγχάνεται επειδή οι συναλλαγές αφορούν **δημόσια κλειδιά**
- Δεν γνωρίζουμε ποια δημόσια κλειδιά ανήκουν σε ποιον
- Κάθε χρήστης δημιουργεί πολλαπλά δημόσια κλειδιά
- Το SHA256² των δεδομένων συναλλαγής ονομάζεται **transaction id (txid)**



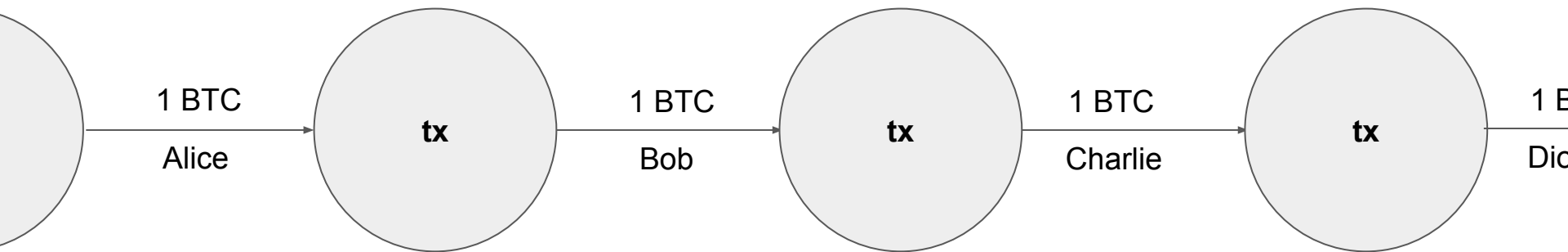




Δημόσιες συναλλαγές στο
blockchain.com

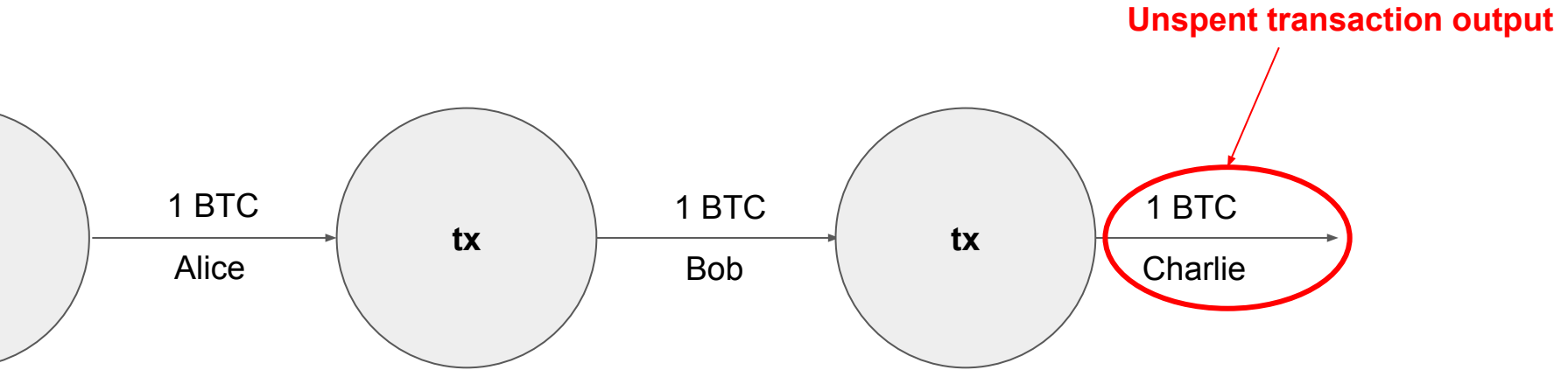
Ο γράφος συναλλαγών

- Οι πληρωμές γίνονται **συνδέοντας** κόμβους συναλλαγών
- Το χρήμα είναι μία **αλυσίδα συναλλαγών**



Αξόδευτα χρήματα

- Τα χρήματα που μπορούν να ξοδευτούν είναι τα **αξόδευτα χρήματα**
- Είναι οι **εξερχόμενες ακμές χωρίς πέρας** από συναλλαγές (utxo)



Πώς ζητάω χρήματα;

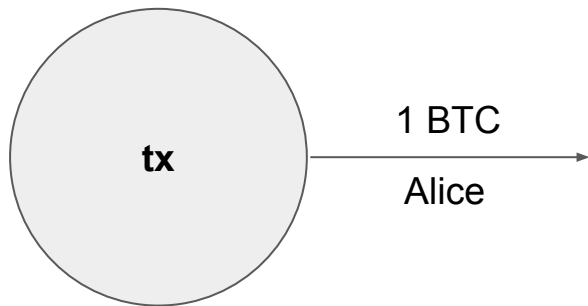
- Παράγω ένα νέο ιδιωτικό κλειδί, αντίστοιχο δημόσιο, και αντίστοιχη διεύθυνση
- Είναι σημαντικό να αλλάζουμε διευθύνσεις για λόγους ανωνυμίας
- **Στέλνω τη διεύθυνση στον πληρωτή** π.χ. μέσω email, FB, QR code κλπ.
- Παρακολουθώ το δίκτυο για κάποια συναλλαγή που με πληρώνει

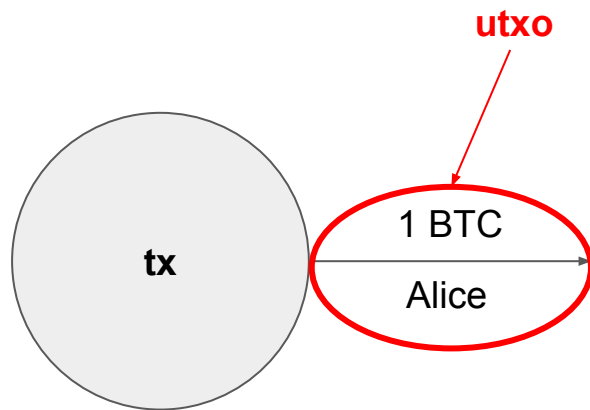
Ποια χρήματα μου ανήκουν;

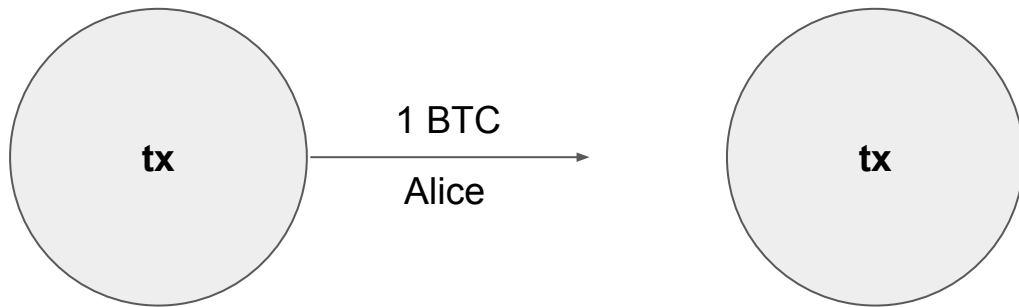
- Όσα βρίσκονται σε UTXO, δηλαδή είναι ακόμη αξόδευτα
 - Διαφορετικά έχω μεταβιβάσει την ιδιοκτησία τους σε κάποιον άλλον
- Στην εξερχόμενη ακμή αναγράφομαι ως ιδιοκτήτης
- Δηλαδή αναγράφεται ένα **δημόσιο** κλειδί για το οποίο κρατώ το ιδιωτικό κλειδί

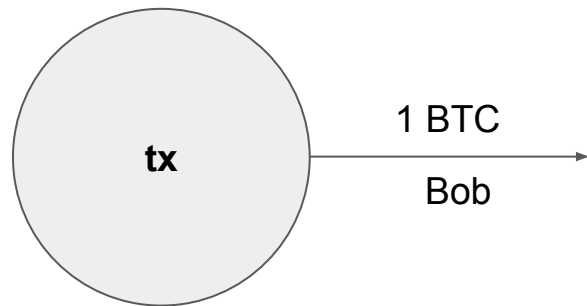
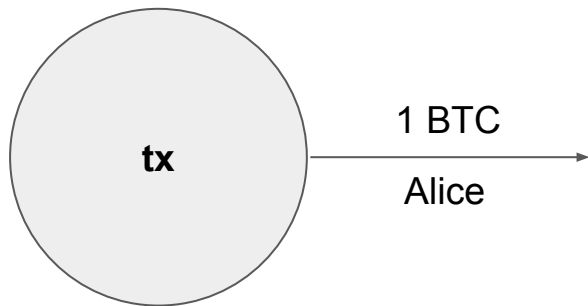
Πώς ξοδεύω χρήματα;

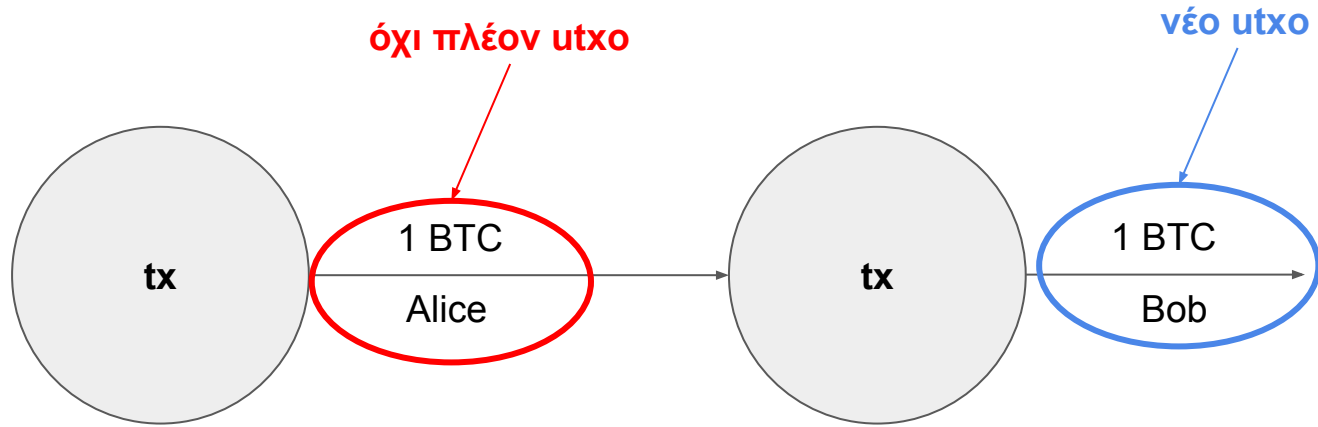
- Βρίσκω μία συναλλαγή που έχει UTXO
- Βεβαιώνομαι ότι **είμαι ο ιδιοκτήτης** της εξερχόμενης ακμής
- Δημιουργώ μία **νέα συναλλαγή**
- **Με μία εισερχόμενη και μία εξερχόμενη ακμή**
- Συνδέω την **εισερχόμενη ακμή της νέας** συναλλαγής με το **παλιό UTXO**
- Πλέον το παλιό utxo δεν είναι πλέον utxo – μόλις ξοδεύτηκε
- Αφήνω την εξερχόμενη ακμή της νέας συναλλαγής ασύνδετη (νέο UTXO)
- Ονομάζω την **αξία** της νέας εξερχόμενης ακμής
- Ονομάζω τον **ιδιοκτήτη** της νέας εξερχόμενης ακμής (δημόσιο κλειδί που προκύπτει από τη διεύθυνση που μου δώθηκε)







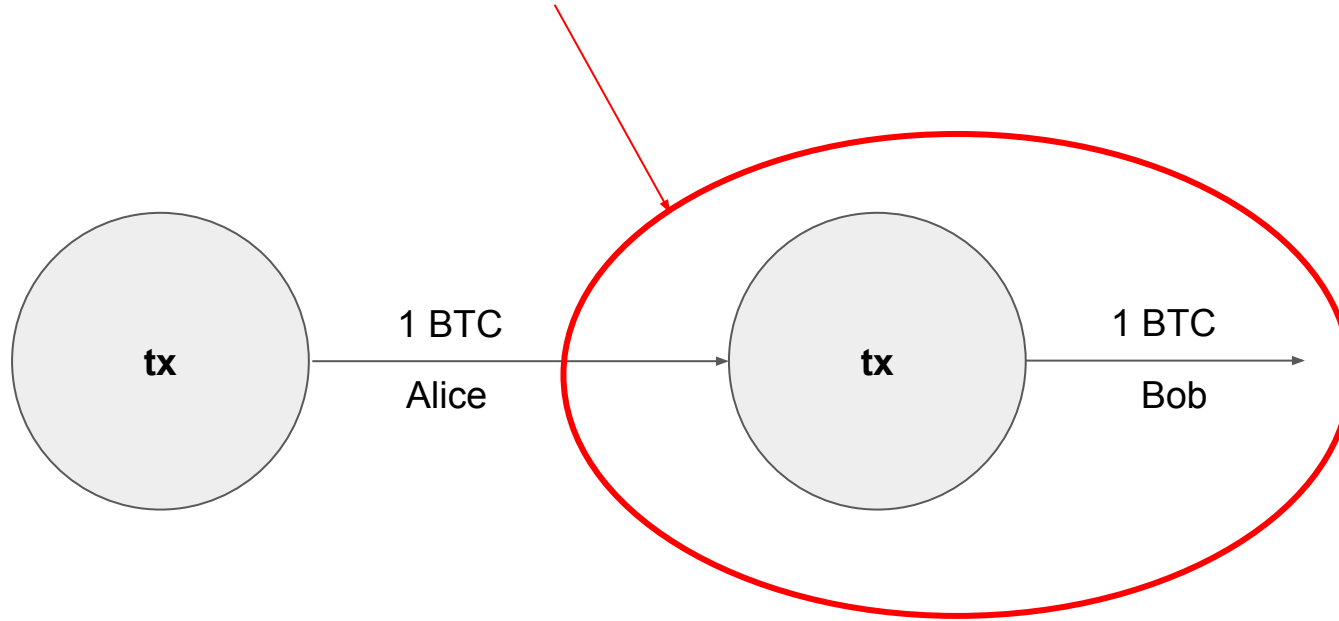




Απόδειξη ιδιοκτησίας

- Υπογράφω ψηφιακά το UTXO που θέλω να ξοδέψω μαζί με τις πληροφορίες της νέας συναλλαγής
- Αυτό εγγυάται ότι είμαι ο πραγματικός ιδιοκτήτης του UTXO
- Η νέα συναλλαγή πρέπει να περιλαμβάνεται στην υπογραφή
- Έτσι εγγυώμαι ότι αδειοδοτώ τον **νέο ιδιοκτήτη** και η υπογραφή μου **δεν μπορεί να παραχαραχθεί** προς λάθος ιδιοκτήτη με απλή αντιγραφή

η Alice υπογράφει

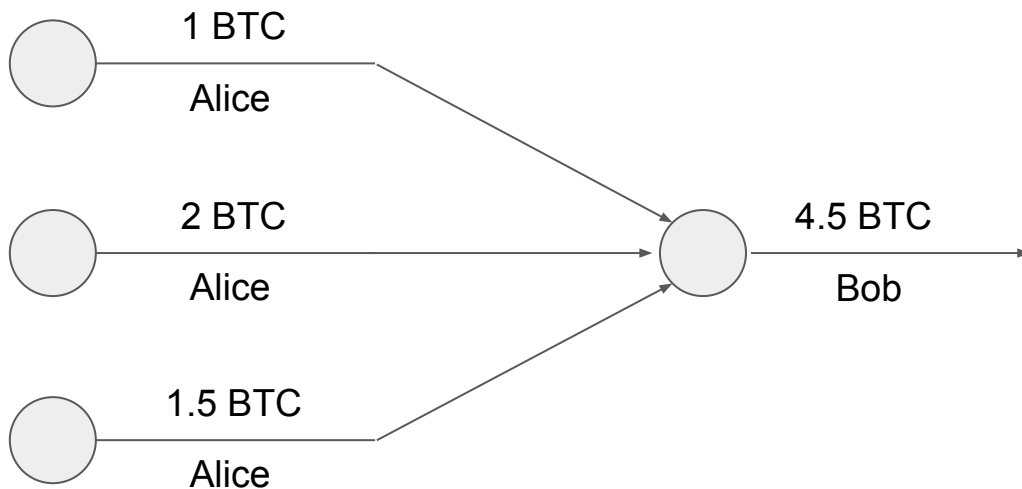


Transaction broadcasting

- **Broadcast:** Όταν δημιουργώ μία συναλλαγή, την στέλνω σε όλους μου τους γείτονες
- **Relay:** Οι γείτονες την στέλνουν στους δικούς τους υπό την προϋπόθεση ότι η συναλλαγή είναι έγκυρη
- Σε λίγο χρόνο, όλο το δίκτυο μαθαίνει για τη συναλλαγή μου

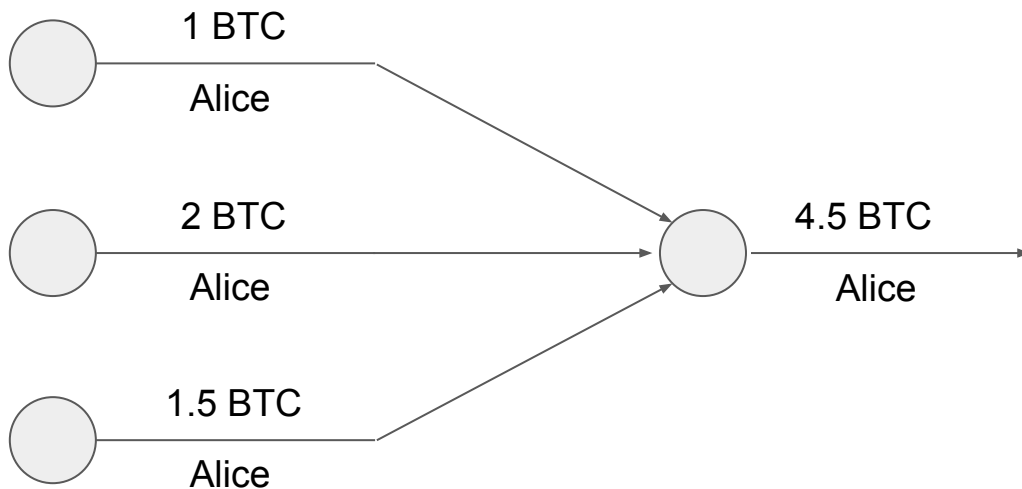
Μία συναλλαγή - πολλές είσοδοι

- Έχω λάβει χρήματα με πολλές συναλλαγές (πολλαπλά UTXOs μου ανήκουν)
- Θέλω να ξοδέψω όλα τα χρήματα μαζί
- Δημιουργώ μία συναλλαγή με πολλές εισόδους και μία έξοδο



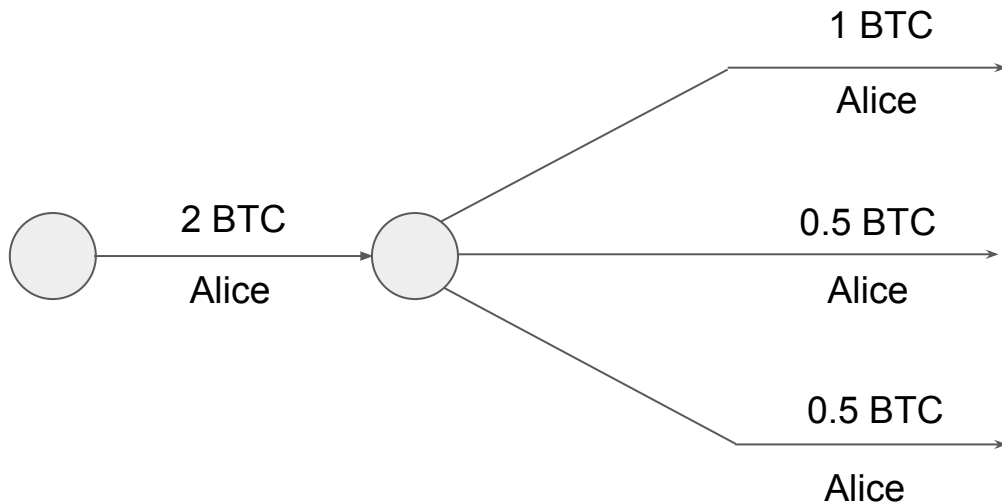
Μία συναλλαγή - πολλές είσοδοι

- Επίσης χρήσιμο αν θέλω να συνδυάσω τα χρήματά μου σε μία διεύθυνση
- Ενώνω τα UTXOs μου μέσω μίας συναλλαγής προς τον εαυτό μου



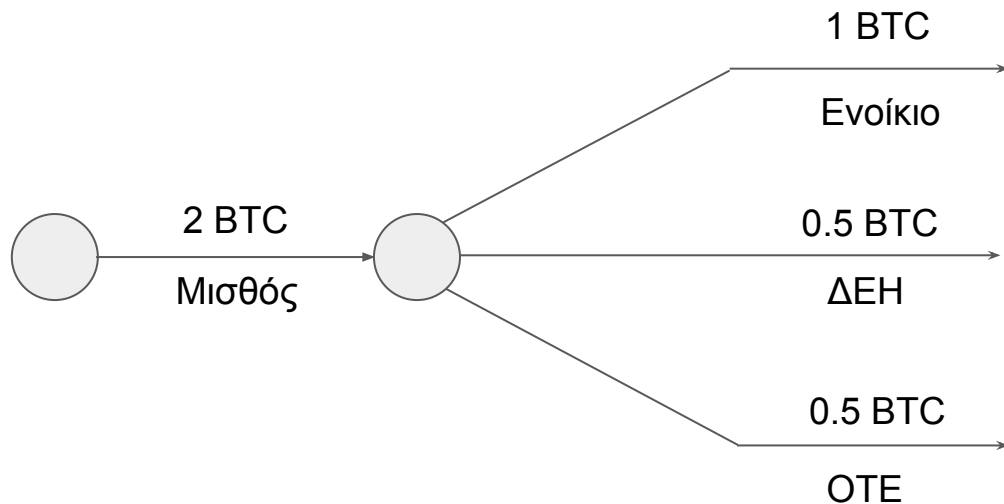
Μία συναλλαγή - πολλές εξόδους

- Έχω μία συναλλαγή με πολλά χρήματα
- Θέλω να τα “σπάσω” σε υποδιαιρέσεις
- Φτιάχνω μία συναλλαγή με μία είσοδο και πολλές εξόδους



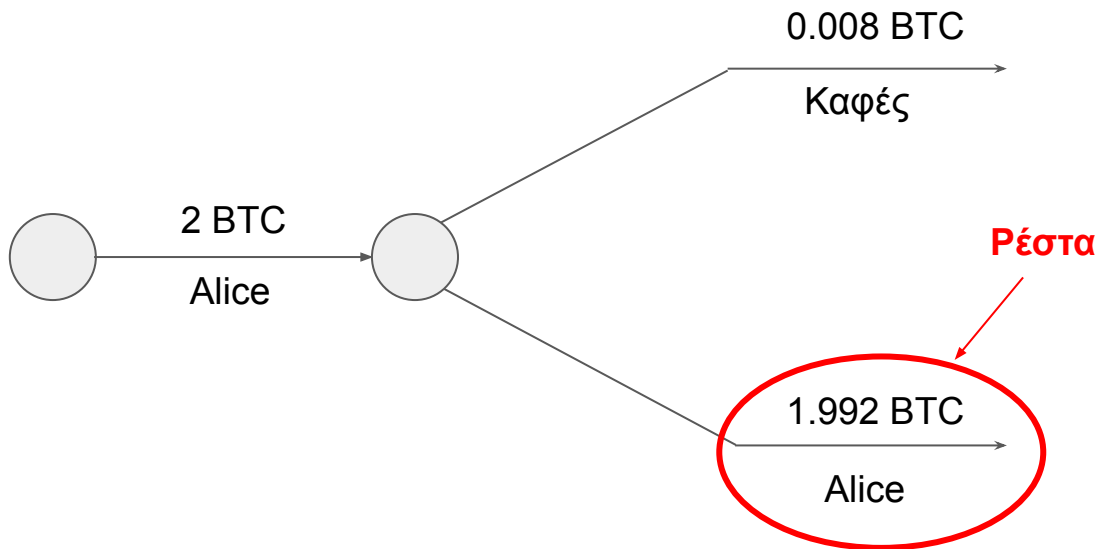
Μία συναλλαγή - πολλές έξοδοι

- Μπορώ να το χρησιμοποιήσω για να κάνω πολλαπλές πληρωμές



Μία συναλλαγή - πολλές έξοδοι

- ...ή για μία μικρή πληρωμή και να κρατήσω τα **ρέστα (change)**
- Τα ρέστα τα δίνω εγώ στον εαυτό μου ως υίτχο, δεν περιμένω από τον πωλητή



Αρχή διατήρησης

$\forall tx \in txs:$

$$\sum_{i \in in(tx)} w(i) \geq \sum_{o \in out(tx)} w(o)$$

Αρχή διατήρησης

Όλες οι συναλλαγές του κόσμου

$$\forall tx \in txs:$$
$$\sum_{i \in in(tx)} w(i) \geq \sum_{o \in out(tx)} w(o)$$

Αξία εισόδου

Αξία εξόδου

The diagram illustrates the conservation principle in a blockchain context. It features the equation $\sum_{i \in in(tx)} w(i) \geq \sum_{o \in out(tx)} w(o)$ for all transactions tx . Annotations include: a red circle around $tx \in txs$ with the label "Όλες οι συναλλαγές του κόσμου" (All transactions of the world); an orange circle around the left sum $\sum_{i \in in(tx)} w(i)$ with the label "Αξία εισόδου" (Input value); and a blue circle around the right sum $\sum_{o \in out(tx)} w(o)$ with the label "Αξία εξόδου" (Output value).

Το σύνολο UTXO

- Το σύνολο όλων των UTXOs του δικτύου είναι σημαντικό
- Δείχνει σε όλους ποια χρήματα μπορούν να ξοδευτούν
- Ό,τι δεν είναι στο UTXO δεν μπορεί να ξοδευτεί
- Γι' αυτό το λόγο, κάθε κόμβος του bitcoin διατηρεί κάθε στιγμή αυτό που πιστεύει ότι είναι το **έγκυρο UTXO set**

Εγκυρότητα μίας συναλλαγής

- Για να επιβεβαιώσουμε την εγκυρότητα μίας συναλλαγής:
- **Επαγωγικά** γνωρίζουμε κάποιες **ήδη έγκυρες** συναλλαγές
 - Διατηρούμε ένα **έγκυρο UTXO set**
- Επιβεβαιώνουμε ότι ισχύει ο νόμος του Kirchhoff
- Επιβεβαιώνουμε την ψηφιακή υπογραφή
- Επιβεβαιώνουμε ότι οι είσοδοι της νέας συναλλαγής συνδέονται **στο έγκυρο UTXO set** που γνωρίζουμε
 - Αυτό επιβεβαιώνει ότι τα χρήματα ξοδεύονται **ακριβώς μία φορά**
- Ενημερώνουμε το έγκυρο UTXO set:
 - **Αφαιρούμε** τα UTXOs που ξοδεύτηκαν
 - **Προσθέτουμε** τα UTXOs που δημιουργήθηκαν

Πόσα bitcoin έχω;

- Παρατηρώ το δίκτυο για συναλλαγές και διατηρώ ένα έγκυρο UTXO set
- Από το έγκυρο UTXO κρατώ τις ακμές που μου ανήκουν
 - Δηλαδή ακμές στις οποίες αναγράφονται δημόσια κλειδιά για τα οποία κρατώ ιδιωτικά κλειδιά
- Αθροίζω τις αξίες
- Το αποτέλεσμα είναι τα χρήματα στην ιδιοκτησία μου

Πορτοφόλι

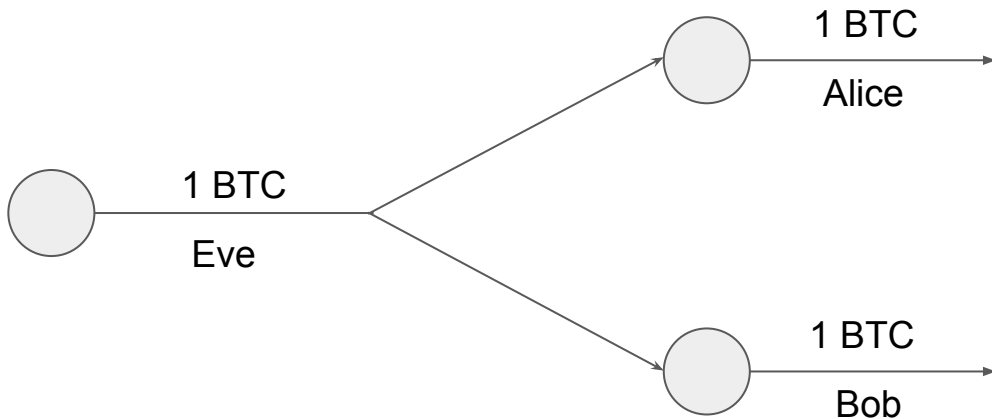
- Ένα **σύνολο ιδιωτικών κλειδιών** bitcoin
- Συνήθως ένα πρόγραμμα
- Τρέχει στον υπολογιστή ή στο κινητό

Double spending

- Τι θα γίνει αν ξοδέψω **το ίδιο UTXO** δύο φορές;
- Η συναλλαγή δεν θα είναι έγκυρη
- Η **πρώτη** συναλλαγή θα είναι έγκυρη
- Η **δεύτερη** συναλλαγή δεν θα είναι έγκυρη
- Αν είχαμε έναν κεντρικό server, αυτό θα ήταν εύκολο...
- Τότε απλώς διατηρούμε ένα σίγουρα έγκυρο UTXO
- Στο p2p δίκτυο του bitcoin μπορεί να καθυστερήσουμε να μάθουμε για κάποια συναλλαγή...
- Μπορεί η Alice να “βλέπει” διαφορετική σειρά συναλλαγών από τον Bob

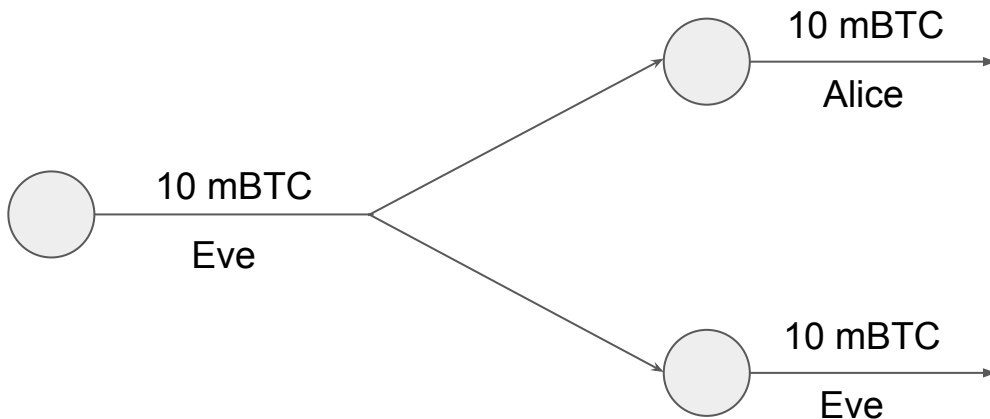
Double spending

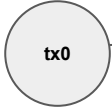
- Δύο συναλλαγές που ξοδεύουν το ίδιο output ονομάζονται **double spend**
- Ο νόμος του Kirchhoff ισχύει για κάθε συναλλαγή
- Όλες οι υπογραφές είναι έγκυρες



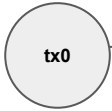
Double spending attack

- Η Eve αγοράζει έναν καφέ από την Alice
- Ταυτόχρονα κάνει double spend προς τον εαυτό της
- Παίρνει τον καφέ και φεύγει
- Η Alice μαθαίνει για το double spend αργότερα

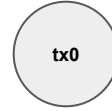




Eve

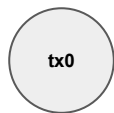


Eve

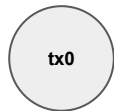


Eve

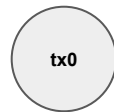




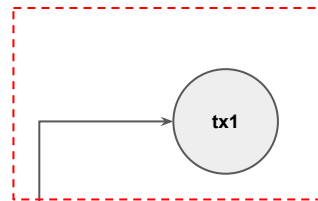
Eve



Eve

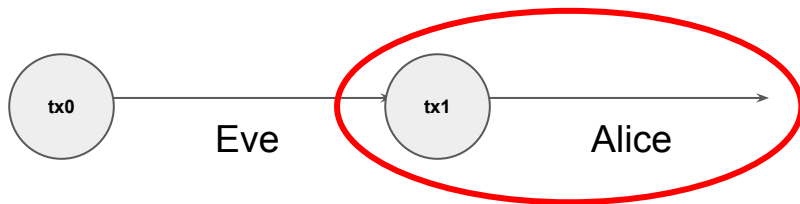


Eve

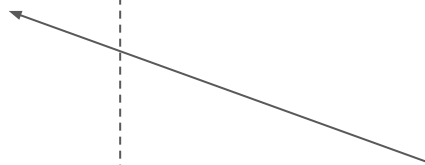


σ_1, m_1

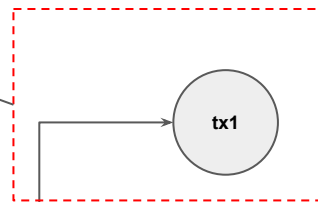




Alice



σ_1, m_1

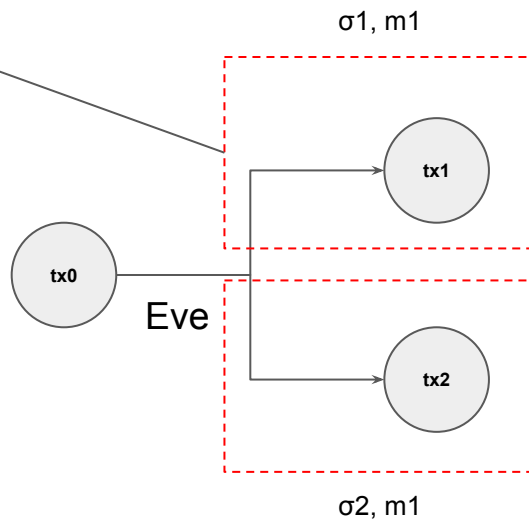
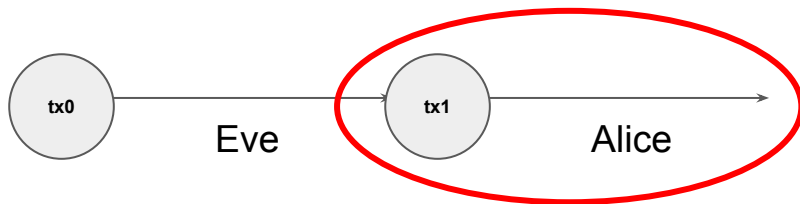


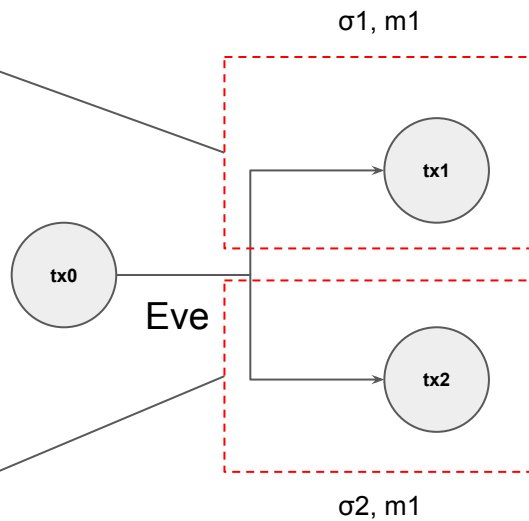
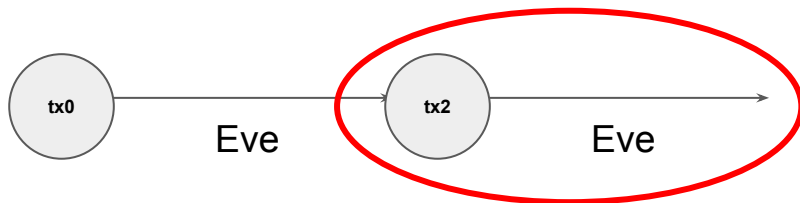
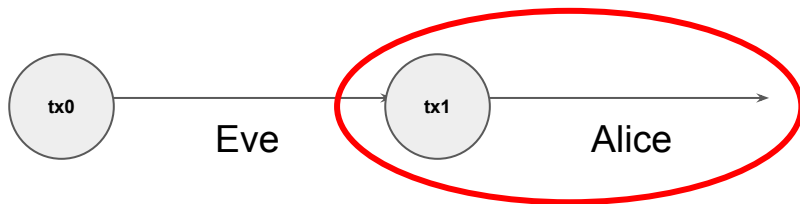
Eve

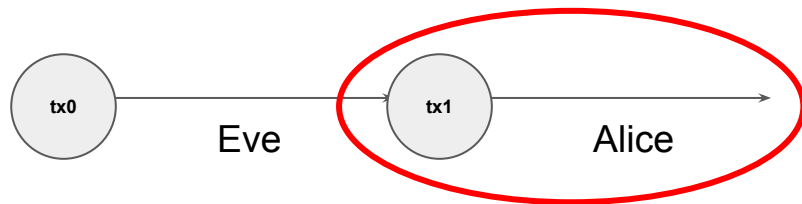


Eve



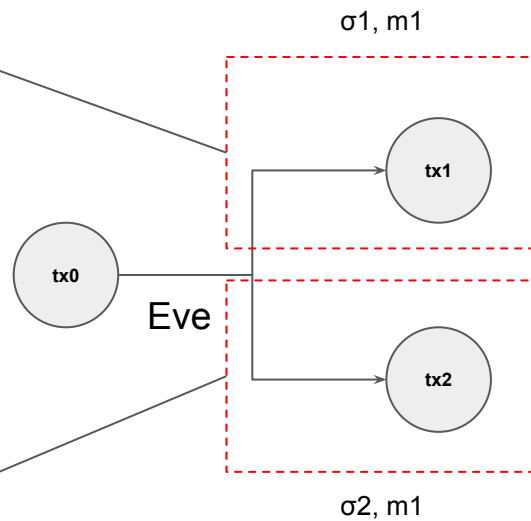
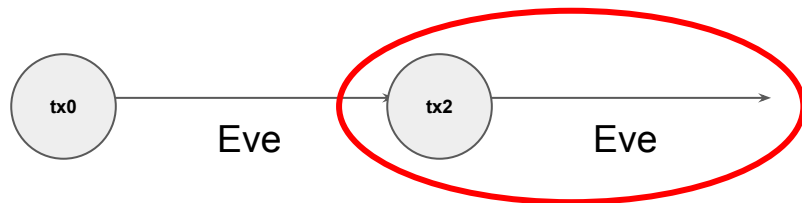


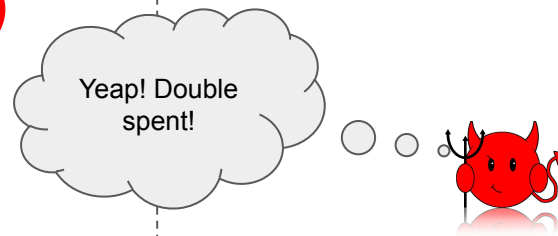
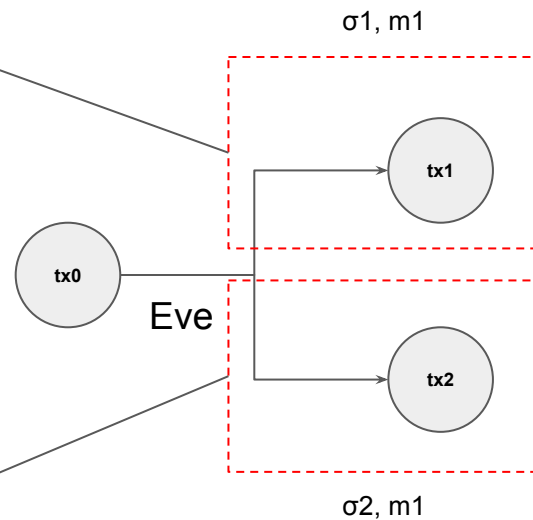
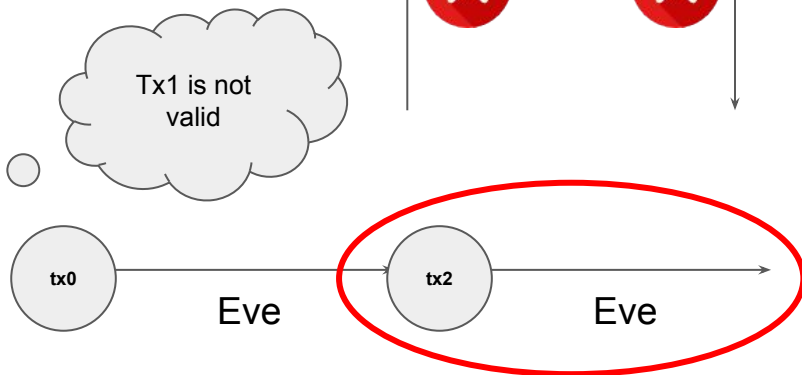
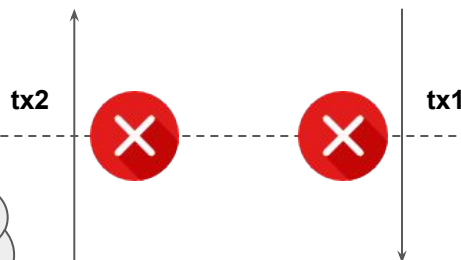
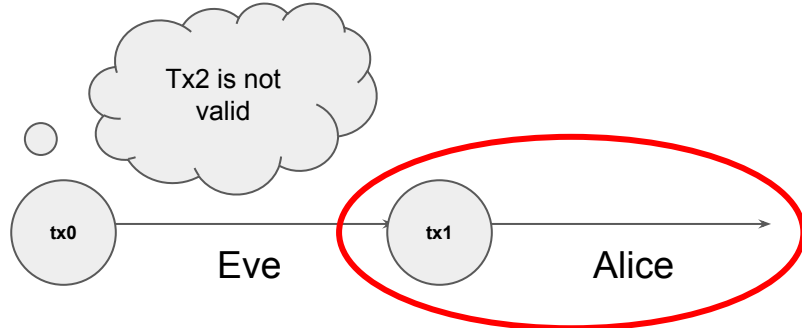




$tx2$

$tx1$



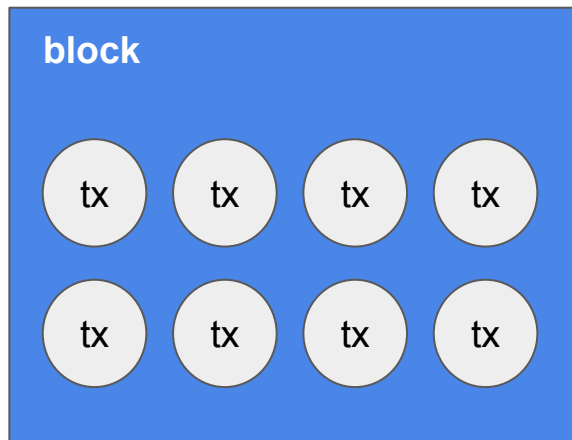


Το βέλος του χρόνου

- Θέλουμε να βάλουμε τις συναλλαγές σε μία σειρά
- Πρέπει να μπορούμε να απαντήσουμε στην ερώτηση: Η συναλλαγή A έγινε πριν την συναλλαγή B;
- Η απάντηση πρέπει να είναι **κοινή για όλους στο δίκτυο**
- Η συμφωνία σε μία κοινή αλήθεια όσο αφορά την ακολουθία συναλλαγών ονομάζεται **consensus**

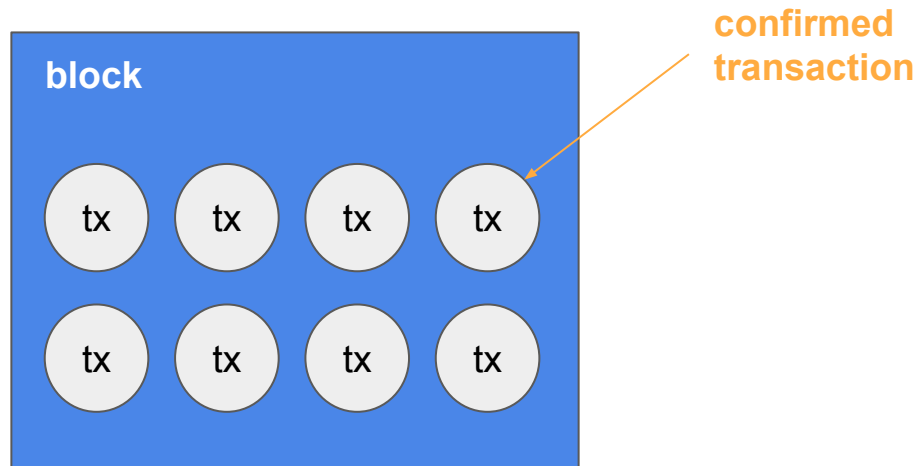
Block

- Συλλέγει πολλά transactions
- Δεν περιέχει double spends, δηλαδή tx που ξοδεύουν το ίδιο output
- Κάθε transaction μπορεί να περιλαμβάνεται **μία φορά** σε ένα block



Block

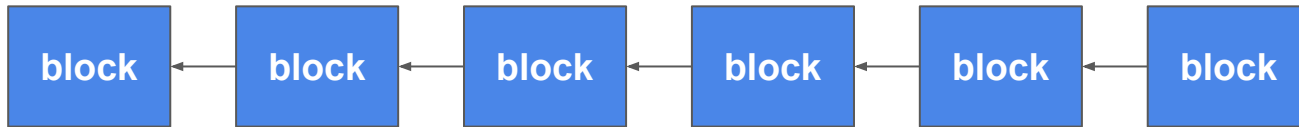
- Το δίκτυο φροντίζει να δημιουργείται καθολικά **ένα block** κάθε **10 λεπτά**
- Το block που δημιουργείται κάθε 10 λεπτά περιλαμβάνει τις **πιο πρόσφατες συναλλαγές** που **δεν υπήρχαν** σε προηγούμενα blocks
- Τα blocks γίνονται **broadcast** και **relay** στο δίκτυο όπως οι συναλλαγές
- Το SHA256 των δεδομένων του block είναι το **block id**
- Μία συναλλαγή που περιλαμβάνεται σε έγκυρο block λέγεται **confirmed**



$$\text{blockid} = \text{SHA256} \left(\begin{array}{|c|} \hline \text{block} \\ \hline \begin{array}{cc} \text{tx} & \text{tx} \\ \text{tx} & \text{tx} \end{array} \\ \hline \end{array} \right)$$

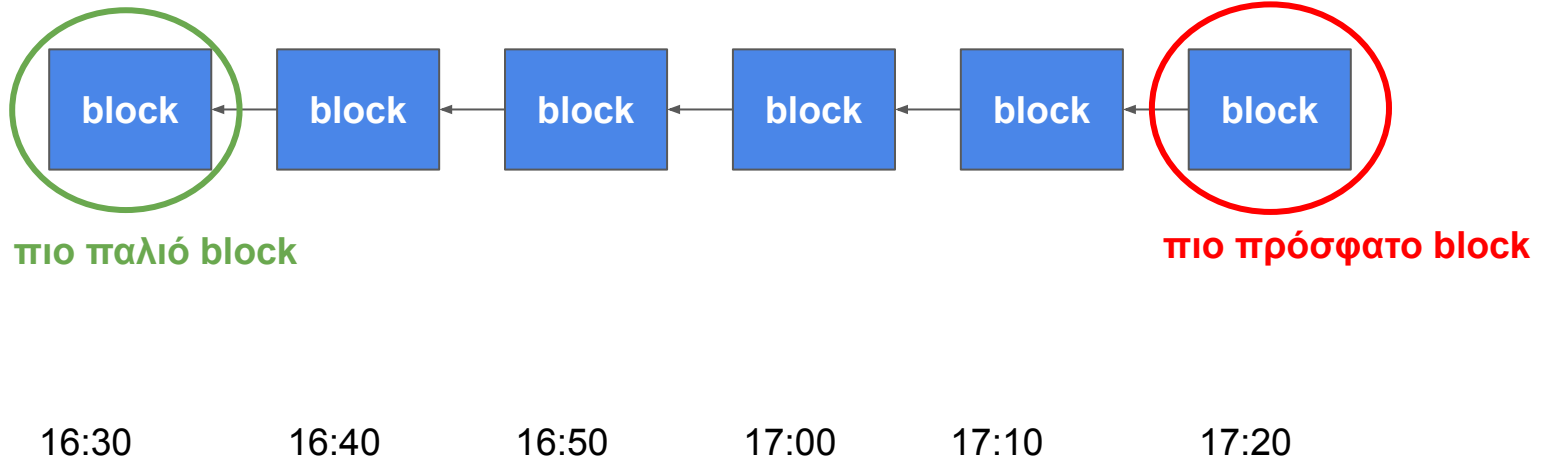
Blockchain

- Κάθε block αναφέρεται στο **προηγούμενο** block
- Περιλαμβάνει ένα δείκτη στο blockid του **πατέρα** του
- Επόμενο block δεν μπορεί να περιέχει double spend προηγούμενου
- Αυτή η συνδεδεμένη λίστα ονομάζεται **blockchain**



Blockchain

- Κάθε block αναφέρεται στο **προηγούμενο** block
- Περιλαμβάνει ένα δείκτη στο blockid του πατέρα του
- Επόμενο block δεν μπορεί να περιέχει double spend προηγούμενου
- Αυτή η συνδεδεμένη λίστα ονομάζεται **blockchain**



Blockchain

- Επιτυγχάνει **consensus**
- Η συναλλαγή A **προηγείται** της συναλλαγής B αν η A **περιλαμβάνεται σε προηγούμενο block** από την B
- Αν θέλουμε να σιγουρευτούμε ότι δεν θα γίνει double spend, πρέπει να περιμένουμε το transaction να γίνει confirm

Blocks στο blockchain.com

Ποιος παράγει τα blocks?

- **Καθένας** μπορεί να παράξει ένα block
- Το σύστημα είναι ελεύθερο στον οποιονδήποτε
- Κάθε block πρέπει να περιέχει μία **απόδειξη εργασίας SHA256²**
- Η απόδειξη εργασίας έχει **δυσκολία** που είναι τέτοια ώστε το **συνολικό δίκτυο** του bitcoin να παράγει **1 block ανά 10 λεπτά σε αναμενόμενη τιμή**

$$E(\text{block generation time}) = 10 \text{ min}$$

Εξόρυξη

- Η διαδικασία της παραγωγής blocks ονομάζεται **εξόρυξη** (mining)
- Υπάρχουν πολλοί bitcoin **miners** που επιχειρούν να εξορύξουν blocks
- Κάθε miner έχει μία **μικρή πιθανότητα** να εξορύξει ένα δεδομένο block
- Όταν ένας miner εξορύξει επιτυχώς ένα block το κάνει **broadcast**
- Οι άλλοι miners το κάνουν **relay**

Αλγόριθμος miner

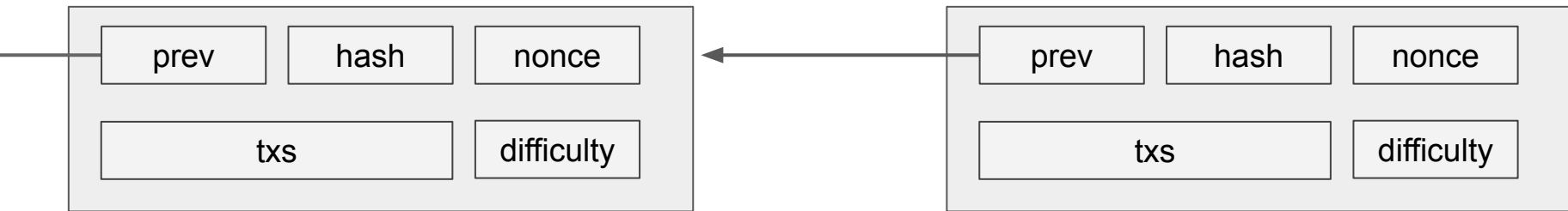
- Παρακολουθούμε το δίκτυο για **συναλλαγές** και **blocks**
- Περιλαμβάνουμε στο **υποψήφιο block** μας:
 - Όλες τις **συναλλαγές** που δεν έχουν εμφανιστεί σε προηγούμενο block που γνωρίζουμε
 - Μία αναφορά στο πιο πρόσφατο block που γνωρίζουμε ως **πατέρα**
- Αναζητούμε **απόδειξη εργασίας**
 - Η απόδειξη εργασίας γίνεται πάνω στον πατέρα και τις συναλλαγές **επιβεβαιώνοντάς** τα
- Αν βρούμε απόδειξη εργασίας κάνουμε **broadcast**
 - Διαφορετικά συνεχίζουμε έως ότου να βρούμε
- Αν μάθουμε ότι κάποιος άλλος miner βρήκε block, πετάμε την προηγούμενη δουλειά μας και συνεχίζουμε να κάνουμε mining πάνω στο πιο πρόσφατο block

Απόδειξη εργασίας bitcoin

$$H(\text{txs} \parallel \text{nonce} \parallel \text{parent-blockid}) < T$$

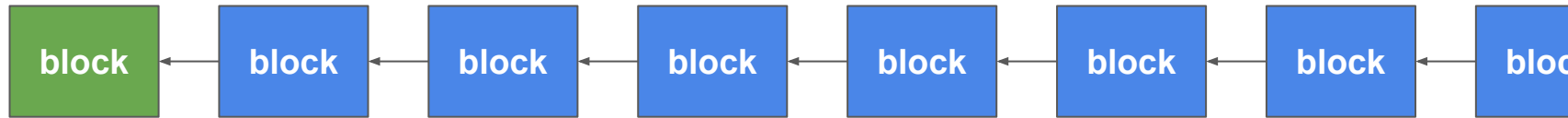
Εγκυρότητα ενός block

- Για να επιβεβαιώσουμε την εγκυρότητα ενός block:
- **Επαγωγικά** γνωρίζουμε **κάποιο ήδη έγκυρο** block
- Επιβεβαιώνουμε ότι το νέο block έχει **πατέρα** το έγκυρο block που γνωρίζουμε
- Επιβεβαιώνουμε την **απόδειξη εργασίας**
- Επιβεβαιώνουμε ότι οι συναλλαγές που περιέχει είναι έγκυρες



Genesis block

- Το **πρώτο** block του blockchain είναι το genesis block
- Είναι **hard-coded** στο bitcoin software
- Κάθε έγκυρο blockchain ξεκινάει από το genesis – είναι η **βάση** της επαγωγής στην επιβεβαίωση εγκυρότητας blocks



genesis block

Genesis block

- Περιλαμβάνει το κείμενο “The Times 03/Jan/2009 Chancellor on brink of second bailout for banks”
- Αυτό αποδεικνύει ότι το block φτιάχτηκε **μετά** τις 3 Ιανουαρίου 2009
- Ξέρουμε επίσης ότι φτιάχτηκε **πριν** τις 3 Ιανουαρίου 2009 επειδή το παρατηρήσαμε στο δίκτυο
- Συνεπώς φτιάχτηκε **στις** 3 Ιανουαρίου 2009
- Η απόσταση ενός block από το genesis ονομάζεται **ύψος (height)**
- Το **block height του genesis** είναι **0**



Eat Out from £5

More than 600 great restaurants, including four Gordon Ramsay favourites from £15

Great collecting ideas today. Perfect for...

Israel prepares to send tanks and troops into Gaza



Chancellor on brink of second bailout for banks

Millions may be needed as banking space tightens

By Andrew Ross
The Chancellor has been warned that the Government will have to provide a second bailout for banks if it does not act quickly. The Bank of England has warned that the banking system is in a state of crisis and that the Government must act to prevent a collapse. The Chancellor has been warned that the Government will have to provide a second bailout for banks if it does not act quickly.

99p



Michael Sheen Frost, Nixon and me



Working mums So that's how she does it



Demos in style The best spots on the planet



Salman Rushdie I won't marry again

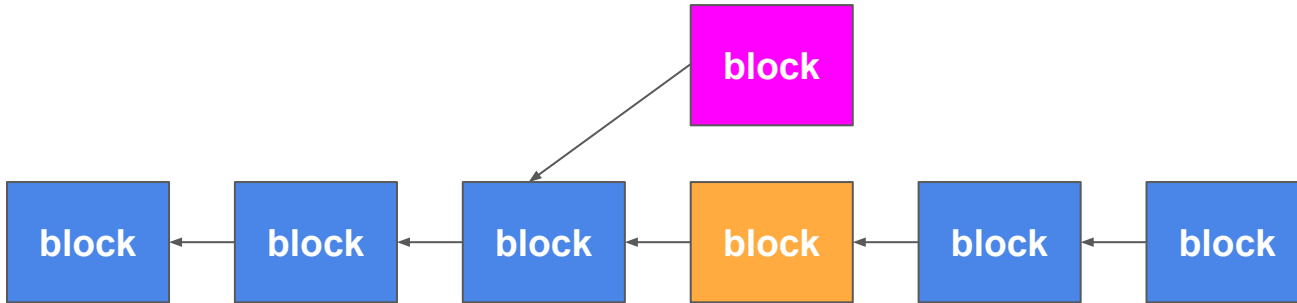


Giant killing? Guide to the FA Cup third round



Blockchain forks

- Κάποιες φορές μπορεί να γίνουν mine 2 έγκυρα blocks ταυτόχρονα
- Αυτό δημιουργεί ένα **blockchain fork**

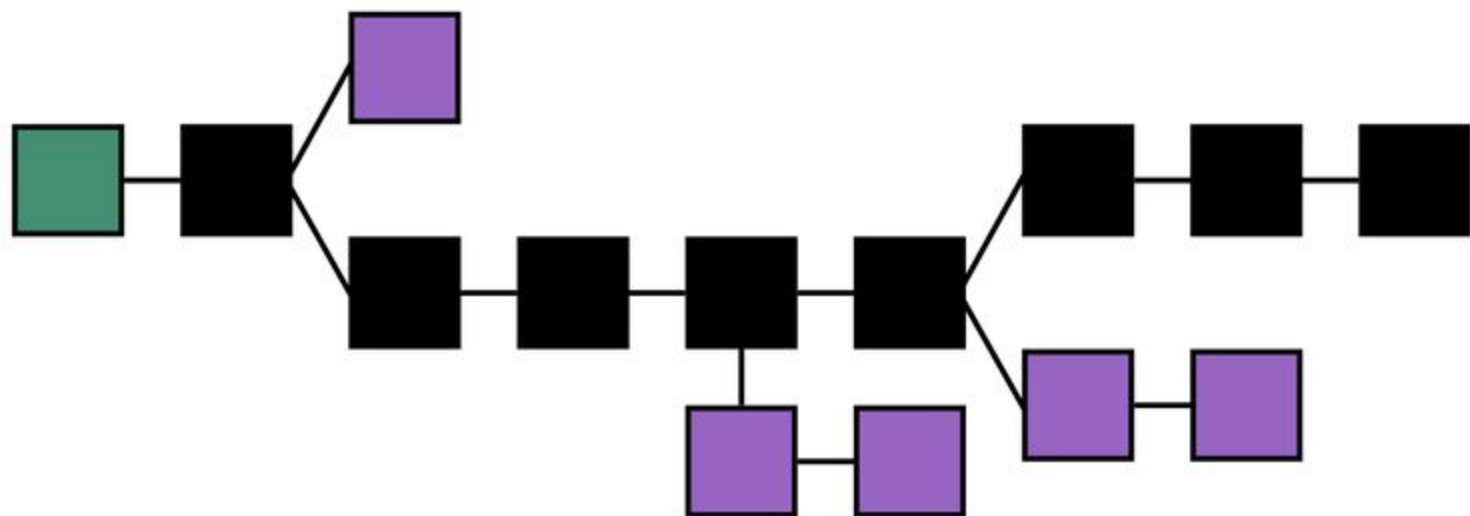


Blockchain fork

- Το blockchain fork είναι πρόβλημα διότι δεν μας επιτρέπει πια να έχουμε βέλος του χρόνου
- Επιστρέφουμε στο ίδιο πρόβλημα που είχαμε με τις συναλλαγές
- Ποιο από τα δύο blocks είναι **το πιο πρόσφατο έγκυρο block**?
- Τι γίνεται αν τα δύο αντίπαλα blocks περιλαμβάνουν **double spends**?

Αλγόριθμος επίλυσης αντίπαλων blockchains

- Παρατηρούμε δύο αντίπαλα blockchains στο δίκτυο
- Το έγκυρο blockchain είναι το blockchain με **το μέγιστο ύψος**
- Αν δύο αντίπαλα blockchains έχουν το ίδιο ύψος, τότε επιλέγουμε κάποιο **αυθαίρετα**
- Το block που επιλέγουμε ως miners είναι αυτό πάνω στο οποίο κάνουμε εξόρυξη
- Το block που επιλέγουμε ως χρήστες είναι αυτό που εμπιστευόμαστε για transaction confirmation



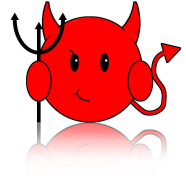
Double spending



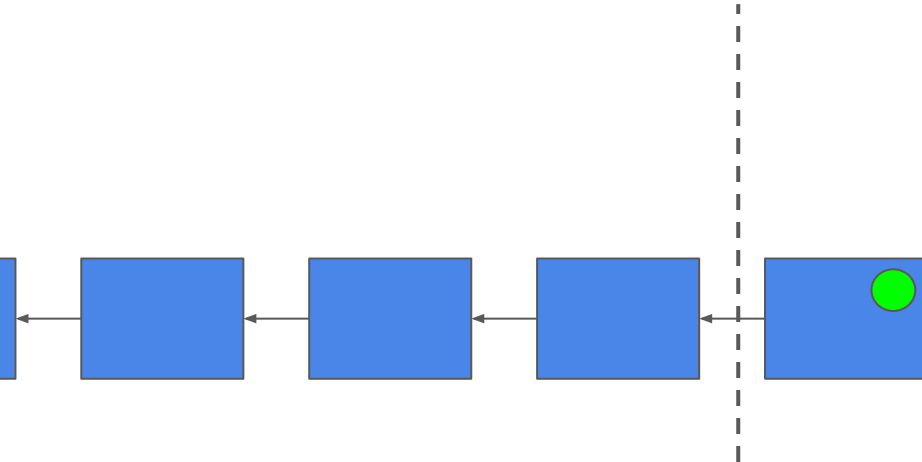
Double spending



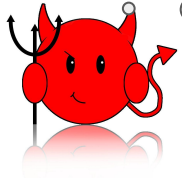
Double spending



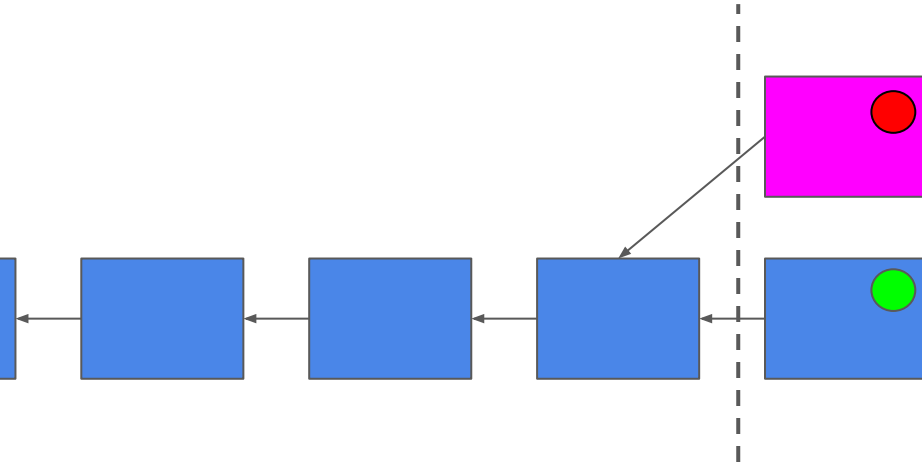
101
010



Double spending



Time to double
spent

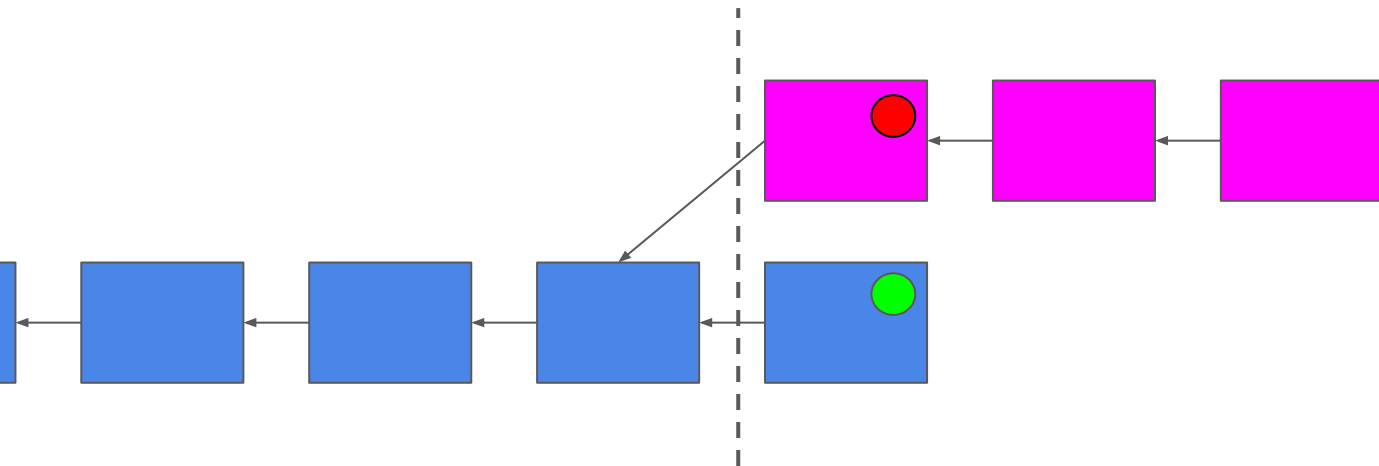


Double spending



Nice!

Wait, what!
I should have waited
for confirmations...

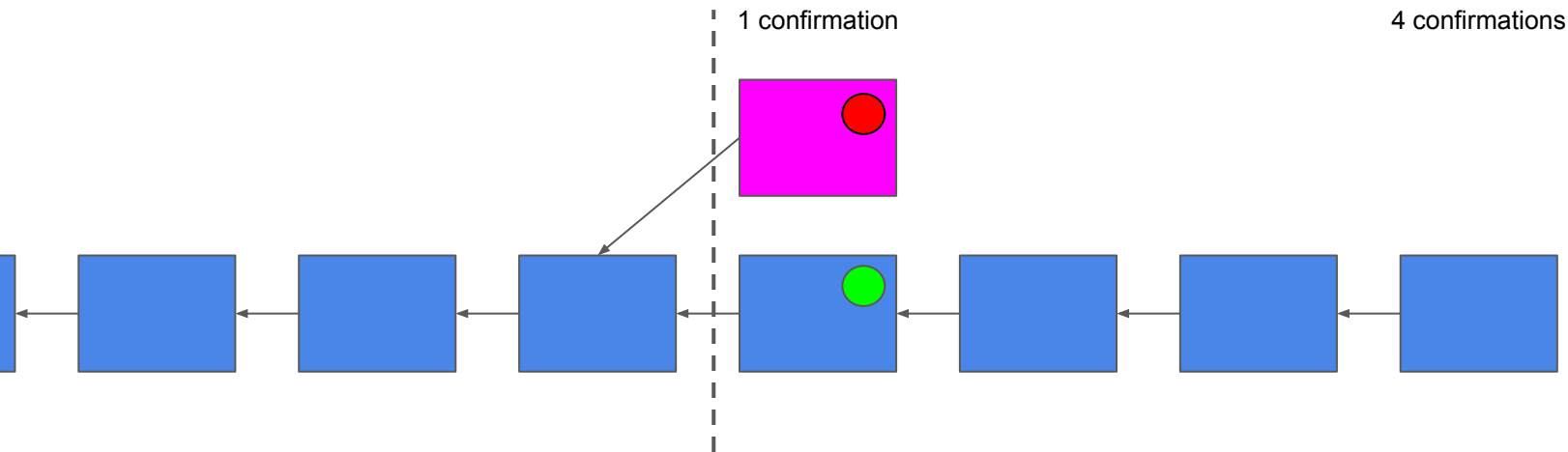


Double spending

Ok, Eve paid
and I see 4
confirmations

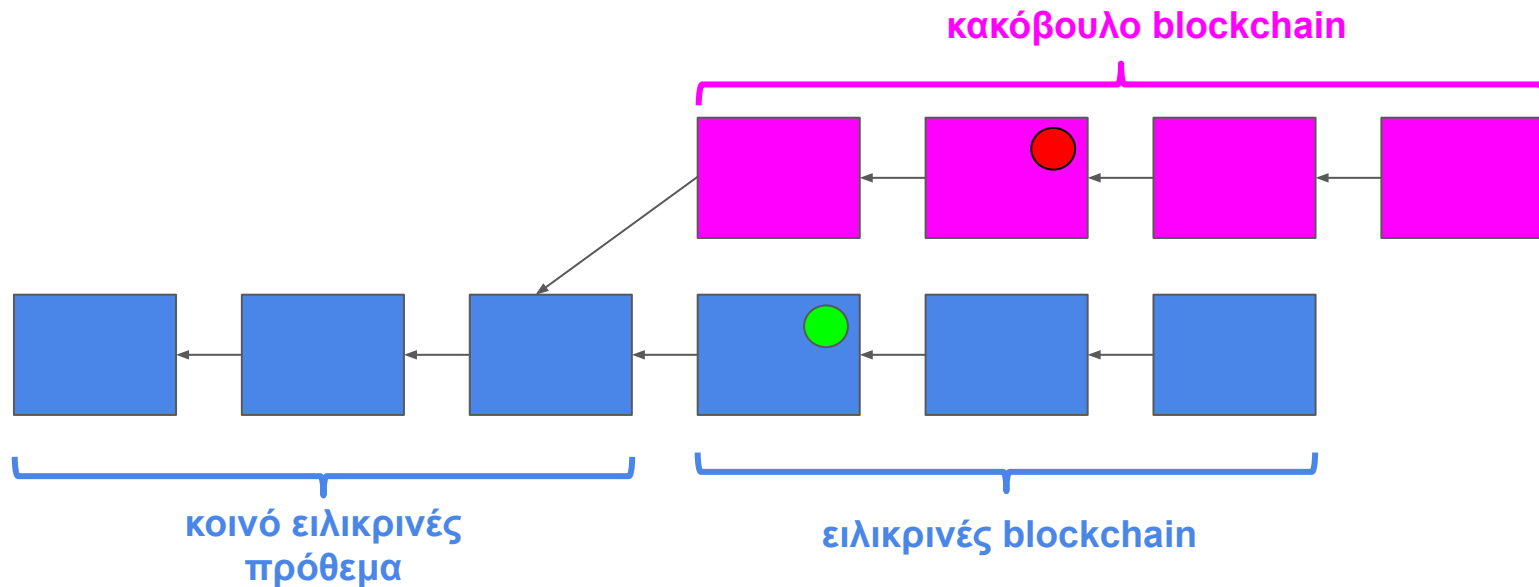


101
010



Double spending

- Για να κάνω double spend πρέπει να παράξω ένα κακόβουλο **παράλληλο blockchain** μεγαλύτερο ή ίσο με το ειλικρινές



Δυσκολία του double spending

- Το double spending απαιτεί μεγάλη υπολογιστική δύναμη
- Ο κακόβουλος θα πρέπει να κατέχει μεγαλύτερη υπολογιστική δύναμη από το υπόλοιπο δίκτυο
- Διαφορετικά η πιθανότητα να μπορεί να συνεχίζει να επεκτείνει το blockchain μειώνεται **εκθετικά** όσο το ειλικρινές blockchain μεγαλώνει
- Μπορεί όμως να το πετύχει αν ελέγχει το 51% της δύναμης CPU του δικτύου
- Αυτό ονομάζεται **51%-attack**

Τι μπορεί να πετύχει ένας κακός miner;

- Μπορεί να κάνει double spending;
 - ?
- Μπορεί να απαγορεύσει χρήματα από το να ξοδευτούν;
 - ?
- Μπορεί να ξοδέψει τα δικά μας χρήματα;
 - ?

Τι μπορεί να πετύχει ένας κακός miner;

- Μπορεί να κάνει double spending;
 - Ναι – φτιάχνει ένα παράλληλο blockchain που περιλαμβάνει την συναλλαγή
- Μπορεί να απαγορεύσει χρήματα από το να ξοδευτούν;
 - Ναι – φτιάχνει ένα παράλληλο blockchain που δεν περιλαμβάνει την συναλλαγή
- Μπορεί να ξοδέψει τα δικά μας χρήματα;
 - Όχι – δεν έχει τα ιδιωτικά κλειδιά μας!

Smart Contracts

- First proposed by Nick Szabo in 1994
- A computerized transaction protocol that executes the terms of a contract
- A set of promises, specified in digital form, including protocols within which the parties perform on these promises.

Smart Contracts

- Define the rules and penalties around an agreement and automatically enforce those obligations
- Many kinds of contractual clauses may be made partially or fully self-executing, self-enforcing, or both
- Minimize the need for trusted intermediaries
- On blockchain: General purpose computation

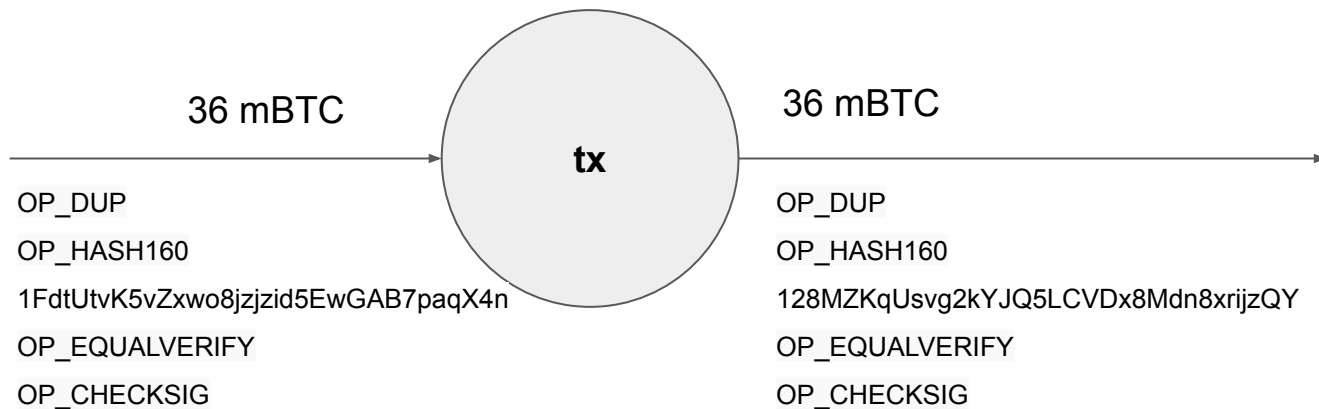
Bitcoin script: The original smart contracts

- People talk about Smart Contracts in Ethereum
- The original Smart Contract language is bitcoin!
- Bitcoin provides a language for **expressing simple smart contracts**
- What can it express?
 - Alice owns some money
 - Alice and Bob own money together
 - Micropayments - continuous transfer of value

Bitcoin script: Encumbrances

- The owner of an edge on the bitcoin tx graph is **not** just bitcoin address!
- It is a **computer program** which decides whether the edge can be spent
- It is written bitcoin script
- This program is called a **scriptPubKey**
- This is the program **the verifier runs**
- This allows us to express more **complicated ownerships**

Bitcoin script



Bitcoin script

- The script runs on a **stack machine**
- It contains **simple serial** commands without loops
- It runs on **every network computer** when a **utxo** is spent
- The output of the execution is 0 or 1
- This is part of transaction validation
- If the output is 1, the input is valid and can be spent
- Otherwise the input is not valid
- And the tx is not valid

Bitcoin script

- When a tx spends a UTXO, the creator of the tx has to prove that the script outputs 1 successfully
 - i.e. that the output edge is spent fairly
- For this purpose, it supplies some **parameters** for the scriptPubKey program so that **when the scriptPubKey program runs with these parameters, it outputs 1**
- The execution parameters of scriptPubKey are called **scriptSig**
- These parameters are given as part of **the new tx** which the old UTXO is connected to

Bitcoin script execution

1. We put **the scriptSig parameters** on the stack
2. We run the **commands of scriptPubKey** one by one
3. Each of these commands can **change** the stack
4. We check if the stack ends up with just a 0 or 1 in the end for **failure** or **success**

Bitcoin script commands

- Built for Bitcoin (inspired by Forth)
- Simple, compact
- Support for cryptography
- Stack-based
- No looping (Not Turing-complete!)
- Time/memory usage bound by program size
- Stateless

Bitcoin script commands

256 opcodes total (15 disabled, 75 reserved)

- Arithmetic
- If/then
- Logic/data handling
- Hashes
- Signature verification
- Multi-signature verification

Bitcoin script commands

- **OP_DUP:**
Duplicates the top element of the stack and put it on the top
- **OP_HASH160:**
Replaces the top element of the stack x with RIPEMD160(SHA256(x))
- **OP_HASH256:**
Replaces the top element of the stack x with SHA256(x)
- **OP_EQUAL:**
Replaces the top two elements of the stack x and y with 1 if $x==y$ and with 0 otherwise

Commands of Bitcoin script

- **OP_VERIFY:**
Removes the top element of the stack. If the element is 1 the program continues. Otherwise the program fails and the execution stops
- **OP_CHECKSIG:**
It takes from the stack a public key and a signature. It checks that the signature has been made on the new transaction and with that particular public key.
- **Constant:**
Adds a constant at the top of the stack

Pay-to-pubkey (p2pk)

- The simplest smart contract
- And the first ever written
- Expresses the notion that some money *rightfully belongs to* an owner

Pay-to-pubkey

scriptPubKey:

045a5f526dfe5d5995bf95f12
OP_CHECKSIG

scriptSig:

υπογραφή σ

Pay-to-pubkey

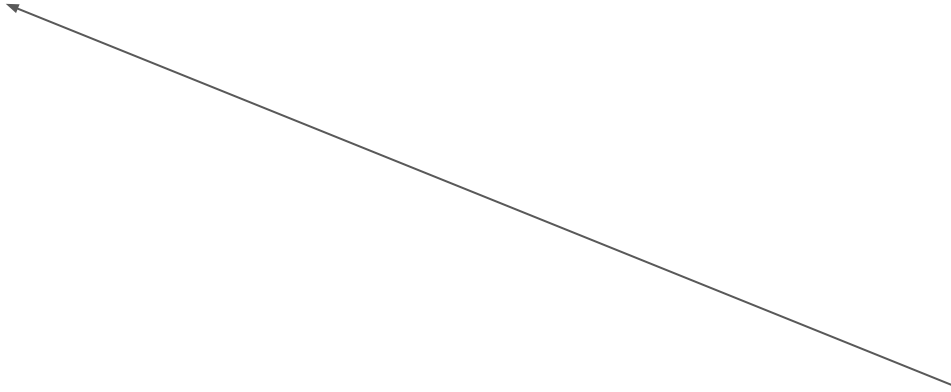
σ

scriptPubKey:

045a5f526dfe5d5995bf95f12
OP_CHECKSIG

scriptSig:

υπογραφή σ



Pay-to-pubkey

σ

scriptPubKey:

→ 045a5f526dfe5d5995bf95f12
OP_CHECKSIG

scriptSig:

υπογραφή σ

Pay-to-pubkey

045a5f526dfe5d5995bf95f12

σ

scriptPubKey:

045a5f526dfe5d5995bf95f12

OP_CHECKSIG

scriptSig:

υπογραφή σ

Pay-to-pubkey

045a5f526dfe5d5995bf95f12

σ

scriptPubKey:

045a5f526dfe5d5995bf95f12

→ OP_CHECKSIG

scriptSig:

υπογραφή σ

Pay-to-pubkey

1



**The transaction
completed successfully**

scriptPubKey:

045a5f526dfe5d5995bf95f12
OP_CHECKSIG

scriptSig:

υπογραφή σ

Pay-to-pubkey-hash

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

pubKey
υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

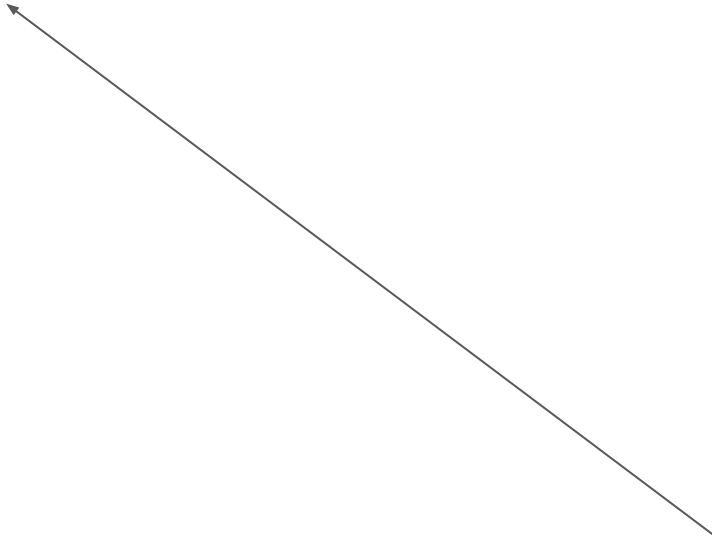
OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ



Pay-to-pubkey-hash

pubKey

υπογραφή σ

scriptPubKey:

→ OP_DUP
OP_HASH160
1FdtUtvK5vZxwo8jzjzid5Ew
OP_EQUALVERIFY
OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

pubKey

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

pubKey

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

→ OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

$H(\text{pubKey})$

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

$H(\text{pubKey})$

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

→ 1FdtUtvK5vZxwo8jzjzid5Ew

OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

1FdtUtvK5vZxwo8jzjzid5Ew

$H(\text{pubKey})$

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

1FdtUtvK5vZxwo8jzjzid5Ew

$H(\text{pubKey})$

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

→ OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

1

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

→ OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

→ OP_EQUALVERIFY

OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

pubKey

υπογραφή σ

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

OP_EQUALVERIFY

→ OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

Pay-to-pubkey-hash

1

The transaction
completed successfully

scriptPubKey:

OP_DUP

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew

OP_EQUALVERIFY

→ OP_CHECKSIG

scriptSig:

pubKey

υπογραφή σ

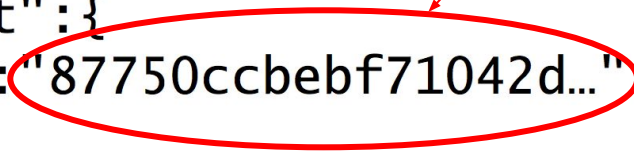
Pay-to-pubkey-hash

- Η πλειονότητα των πληρωμών στο bitcoin σήμερα είναι Pay-to-pubkey-hash
- Το Pay-to-pubkey χρησιμοποιήθηκε στην αρχή
- Πλεονέκτημα του pay-to-pubkey-hash:
- Αμύνεται στο “σπάσιμο” της EC κρυπτογραφίας
- Τα δημόσια κλειδιά δεν αποκαλύπτονται έως ότου έρθει η ώρα να ξοδέψουμε
- Όταν ξοδέψουμε, ο αντίπαλος μπορεί να προσπαθήσει να βρει το ιδιωτικό κλειδί και να κάνει double spend
- Έχει 10 λεπτά έως ότου γίνουμε confirm σε block
- Αν η EC κρυπτογραφία σπάει, δύσκολα σπάει σε 10 λεπτά


```
{
  "hash": "96f5e5394726ca5...",
  "ver": 1,
  "in": [{
    "prev_out": {
      "hash": "87750ccbebf71042d...",
      "n": 0
    },
    "scriptSig": "30440397d0c2... 49d0c04a7e52..."
  }],
  "out": [{
    "value": "0.71430000",
    "scriptPubKey": "OP_DUP OP_HASH160
99fa78c49d99f58c8dd... OP_EQUALVERIFY
OP_CHECKSIG"
  }]
}
```

```
{
  "hash": "96f5e5394726ca5...",
  "ver": 1,
  "in": [{
    "prev_out": {
      "hash": "87750ccbebf71042d...",
      "n": 0
    },
    "scriptSig": "30440397d0c2... 49d0c04a7e52..."
  }],
  "out": [{
    "value": "0.71430000",
    "scriptPubKey": "OP_DUP OP_HASH160
99fa78c49d99f58c8dd... OP_EQUALVERIFY
OP_CHECKSIG"
  }]
}
```

utxo txid



```
{
  "hash": "96f5e5394726ca5...",
  "ver": 1,
  "in": [{
    "prev_out": {
      "hash": "87750ccbebf71042d...",
      "n": 0
    },
    "scriptSig": "30440397d0c2... 49d0c04a7e52..."
  }],
  "out": [{
    "value": "0.71430000",
    "scriptPubKey": "OP_DUP OP_HASH160
99fa78c49d99f58c8dd... OP_EQUALVERIFY
OP_CHECKSIG"
  }]
}
```

Diagram annotations:

- A purple oval highlights the "hash" field of the first input, with a purple arrow pointing to it labeled "txid".
- A red oval highlights the "hash" field of the previous output, with a red arrow pointing to it labeled "utxo txid".
- A red circle highlights the "n" field of the previous output, with a red arrow pointing to it labeled "utxo index".
- The "scriptSig" field is underlined in blue.
- The "value" field is underlined in orange.
- The "scriptPubKey" field is underlined in green.

Ποιος μπορεί να ξοδέψει αυτό το script?

scriptPubKey:

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew...

OP_EQUAL

Ποιος μπορεί να ξοδέψει αυτό το script?

scriptPubKey:

OP_HASH160

1FdtUtvK5vZxwo8jzjzid5Ew...

OP_EQUAL

Λείπει το OP_CHECKSIG! **Οποιοσδήποτε** μπορεί να ξοδέψει αρκεί να ξέρει το public key που αντιστοιχεί στη διεύθυνση.

Ποιος μπορεί να ξοδέψει αυτό το script?

scriptPubKey:

OP_HASH160

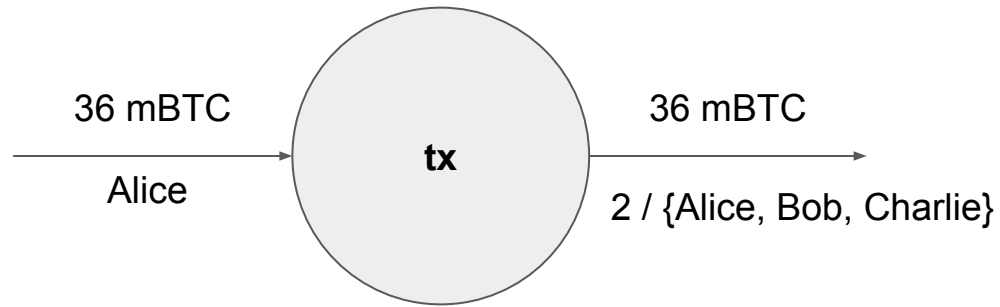
1FdtUtvK5vZxwo8jzjzid5Ew...

OP_EQUAL

Οποιοσδήποτε μπορεί να κάνει **double spend** διότι το public key δημοσιεύεται ως scriptSig στην πρώτη απόπειρα ξοδέματος!

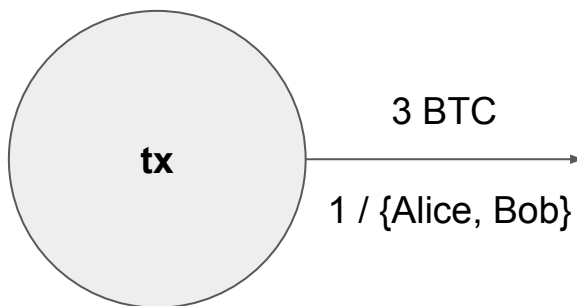
Multisig

- Πιο προχωρημένο scriptPubKey
- Δηλώνει ένα σύνολο n διευθύνσεων bitcoin και δύο αριθμούς m και n
- Δηλώνει ότι μία συναλλαγή μπορεί να ξοδευτεί αν έχουμε οποιεσδήποτε υπογραφές δημιουργημένες με m από αυτά τα n κλειδιά
- $2 / \{Alice, Bob, Charlie\}$: Τη συναλλαγή μπορούν να ξοδέψουν αν συνεργαστούν οποιοιδήποτε 2 (ονομάζεται 2-από-3)
 - $\{Alice, Bob\}$
 - $\{Alice, Charlie\}$
 - $\{Bob, Charlie\}$



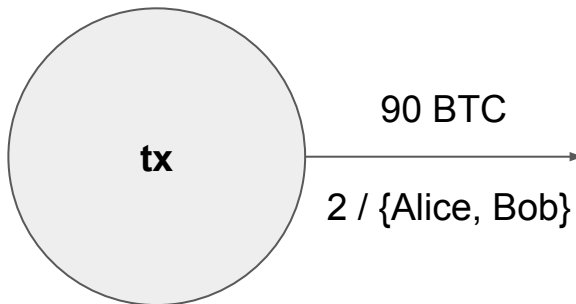
Εφαρμογή 1-από-2: Λογαριασμός όψεως γάμου

- Η Alice και ο Bob παντρεύονται
- Θέλουν να έχουν ένα “κοινό” λογαριασμό
- Δεν χρειάζεται να μοιράζονται κοινά ιδιωτικά κλειδιά
- Δημιουργούν μία 1-από-2 multisig διεύθυνση
- Οποιοσδήποτε από τους δύο μπορεί να ξοδέψει τα χρήματα



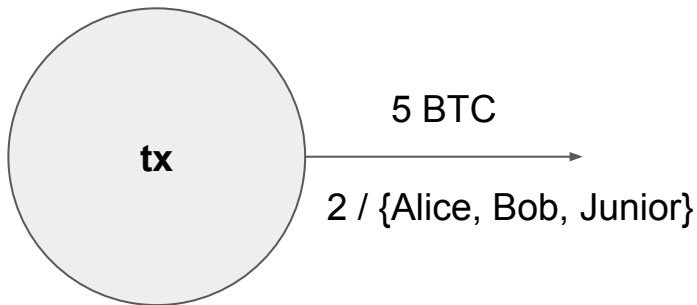
Εφαρμογή 2-από-2: Λογαριασμός ταμιευτηρίου

- Λογαριασμός αποταμίευσης της οικογένειας Alice/Bob
- Απαιτείται η άδεια (= ψηφιακή υπογραφή) και των δύο για να ξοδευτούν χρήματα



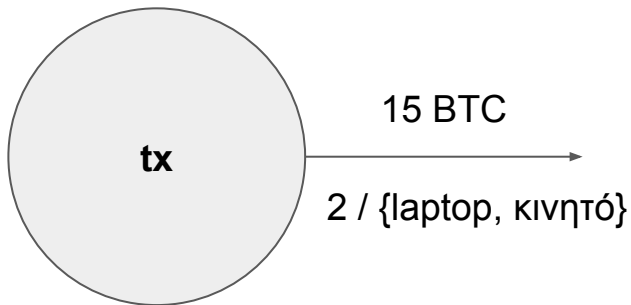
Εφαρμογή 2-από-3: Λογαριασμός παιδιού

- Οι γονείς Alice και Bob φτιάχνουν ένα λογαριασμό bitcoin στον Junior
- Ο Junior μπορεί να ξοδέψει τα χρήματα με την άδεια οποιουδήποτε γονιού
- Οι γονείς πρέπει να συνεργαστούν για να πάρουν τα χρήματα του παιδιού



Εφαρμογή 2-από-2: Second factor authentication

- Κρατάς πολλά χρήματα σε bitcoin
- Έχεις ένα ιδιωτικό κλειδί στο laptop και ένα στο κινητό
- Απαιτούνται 2 υπογραφές για να ξοδευτούν τα χρήματα
- Είναι πολύ πιο δύσκολο κάποιος κακόβουλος να υποκλέψει και τα δύο κλειδιά



Εφαρμογή 2-από-3: Εταιρική διοίκηση

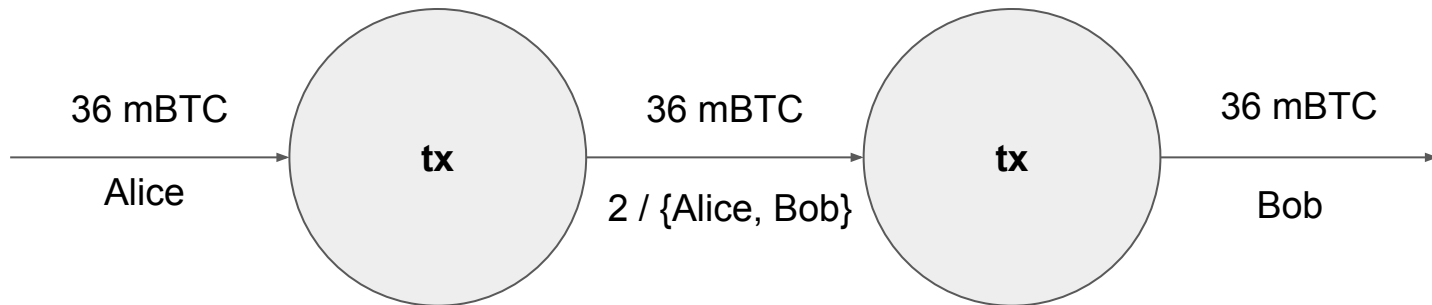
- Τα χρήματα μίας εταιρείας ελέγχονται από 3 μετόχους
- Χρειάζεται πλειοψηφία για να μπορέσουν να τα ξοδέψουν
- Κρατούν τα χρήματα σε μία 2-από-3 multisig διεύθυνση
- Δεν απαιτούνται τράπεζες ή νόμοι για να αδειοδοτηθεί κάθε συναλλαγή
- Το “συμβόλαιο” της πλειοψηφίας δεν μπορεί να παραβιαστεί
- Πρόκειται για ένα smart contract ή **self-enforcing contract**
- Για μεγαλύτερες εταιρείες έχουμε 3-από-5, 5-από-9 κ.ό.κ.

Εφαρμογή 2-από-2

- Η Alice θέλει να αγοράσει ένα προϊόν του Bob
- Η Alice δε θέλει να στείλει πρώτη τα χρήματα
- Ο Bob δε θέλει να στείλει πρώτος το προϊόν

Εφαρμογή 2-από-2

- Η Alice δημιουργεί μία **2-από-2 συναλλαγή** με τα χρήματα
- Ο Bob βλέπει τη συναλλαγή στο blockchain (**proof of payment**)
- Ούτε η Alice ούτε ο Bob μπορούν τώρα να ξοδέψουν τα χρήματα
 - είναι “κλειδωμένα” (locked)
- Ο Bob στέλνει το προϊόν
- Όταν η Alice λάβει το προϊόν, βάζει την μία από τις δύο υπογραφές (**finalize**)
- Ο Bob βάζει τη δεύτερη υπογραφή και λαμβάνει τα χρήματα



Puzzle

Τι κάνει αυτό το script;

OP_HASH256

00000000006fe28c0ab6f1b372c1a6a246ae63f74f931

OP_EQUAL

Puzzle

Τι κάνει αυτό το script;

OP_HASH256

00000000006fe28c0ab6f1b372c1a6a246ae63f74f931

OP_EQUAL

Ανταμείβει όποιον βρει το pre-image αυτού του hash και το δημοσιεύσει ως scriptSig!

Puzzle

Τι κάνει αυτό το script;

OP_HASH256

00000000006fe28c0ab6f1b372c1a6a246ae63f74f931

OP_EQUAL

Ανταμείβει όποιον βρει το pre-image αυτού του hash και το δημοσιεύσει ως scriptSig!

Μπορείτε να [μαντέψετε το preimage](#);

Proof-of-burn

- Το script περιέχει την εντολή OP_RETURN
- **Σταματά** την εκτέλεση του προγράμματος αφήνοντας τη συναλλαγή ως άκυρη
- Οι επόμενες εντολές δεν εκτελούνται – μπορεί να είναι οτιδήποτε
- **Κανείς** δεν μπορεί να την ξοδέψει
- Τι νόημα έχει αυτό;

Proof-of-ownership

- Μπορώ να αποδείξω ότι είχα στην κατοχή μου ένα έγγραφο μία συγκεκριμένη χρονική στιγμή
- Υπολογίζω το $s = \text{SHA256}(\text{έγγραφο})$
- Κάνω μία συναλλαγή αξίας 1 satoshi με το εξής scriptPubKey:

OP_RETURN

s

Proof-of-ownership

- Μπορώ να αποδείξω ότι είχα στην κατοχή μου ένα έγγραφο μία συγκεκριμένη χρονική στιγμή
- Υπολογίζω το $s = \text{SHA256}(\text{έγγραφο} \parallel \text{το-όνομά-μου})$
- Κάνω μία συναλλαγή αξίας 1 satoshi με το εξής scriptPubKey:

OP_RETURN

S ← μέρος του προγράμματος, αλλά δεν εκτελείται ποτέ

Proof-of-ownership

- Μπορώ να αποδείξω ότι είχα στην κατοχή μου ένα έγγραφο μία συγκεκριμένη χρονική στιγμή
- Υπολογίζω το $s = \text{SHA256}(\text{έγγραφο} \parallel \text{το-όνομά-μου})$
- Κάνω μία συναλλαγή αξίας 1 satoshi με το εξής scriptPubKey:

OP_RETURN

s

- Αργότερα δημοσιεύω το έγγραφο με το όνομά μου αποδεικνύοντας ότι το είχα την ημερομηνία που έγινε confirm το tx

A more complicated contract

OP_2DUP

OP_HASH160

BOB_HASH_CONST

OP_EQUALVERIFY

OP_DUP

OP_HASH160

ALICE_HASH_CONST

OP_EQUALVERIFY

OP_SIZE

OP_NIP

16

OP_NUMEQUAL

OP_SWAP

OP_SIZE

OP_NIP

16

OP_NUMEQUAL

OP_NUMEQUAL

OP_IF

ALICE_PUB_KEY

OP_ELSE

BOB_PUB_KEY

OP_END_IF

OP_CHECKSIG

Μάθαμε

- Συναλλαγές, ρέστα
- Γράφος του bitcoin, ακμές, κόμβοι, αξίες, ιδιοκτήτες, utxo, coinbase
- Εξόρυξη, consensus, blockchain, genesis στο bitcoin
- Bitcoin script
- p2pk, p2pkh, multisig

