Lezioni Lincee di Scienze Informatiche

Milano–Bicocca – 23 febbraio 2018

Crittografia a catenelle: teoria e applicazioni delle Blockchain

Alberto Leporati

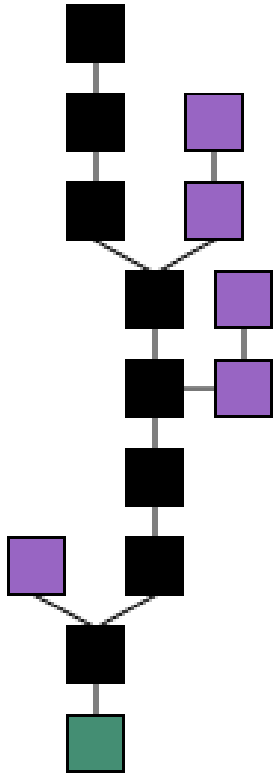
Università degli Studi di Milano – Bicocca

Dip. di Informatica, Sistemistica e Comunicazione (DISCo)

Viale Sarca 336/14 – Milano - Italy

Blockchain, in breve

- Un registro (ledger) decentralizzato, condiviso, pubblico, che registra il possesso di beni digitali
- I nodi contengono transazioni (trasferimenti di proprietà)
- Si parte da un genesis block (in verde)
- Ogni nodo è legati al successivo tramite il calcolo di una funzione hash
- I nodi vengono creati da una rete P2P, i cui membri validano le transazioni attraverso un protocollo di consenso condiviso, basato su una proof-of-work
- Le transazioni sono (pseudo)-anonime
- Possono essere presenti dei nodi orfani (in viola)



(Fonte: Wikipedia)

Bitcoin, in breve

- Crittovaluta proposta da Satoshi Nakamoto nel 2008
- I beni digitali sono i bitcoin (BTC), o loro frazioni
- Il possesso consiste nel conoscere una chiave segreta
- Per trasferire il possesso si usa la Crittografia a chiave pubblica:
  - si indica la chiave pubblica del destinatario
  - si firma la transazione con la propria chiave segreta
- La blockchain registra ogni transazione
  - attualmente (22 feb 2018), 148.5 GB
- Prima soluzione decentralizzata al problema del double spending
- Satoshi Nakamoto = Nick Szabo? + Hal Finney + Wei Dai?

Cos'è la Crittografia?

- Scopo della crittografia: studiare metodi che consentano di memorizzare, elaborare e trasmettere informazioni in presenza di agenti ostili
- Il nome deriva da due parole greche:

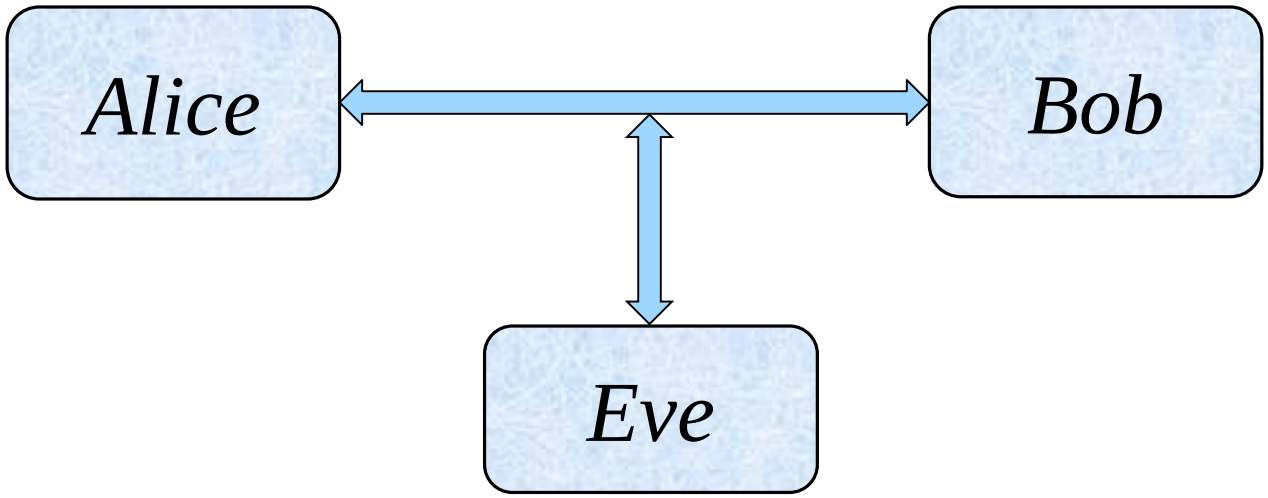
  κρυπτός (*kryptós*): "nascosto"

  γραφία (*graphía*) : "scrittura"
- Da non confondere con la steganografia!
  - messaggi nascosti in foto/audio/video
  - watermarking e fingerprinting
- Arte antica, e scienza moderna:
  - fino al 1600 – 1700: sostituzioni monoalfabetiche
  - sostituzioni polialfabetiche e sistemi simmetrici fino al 1975 – 1976
  - dal 1976: crittografia a chiave pubblica e moderna

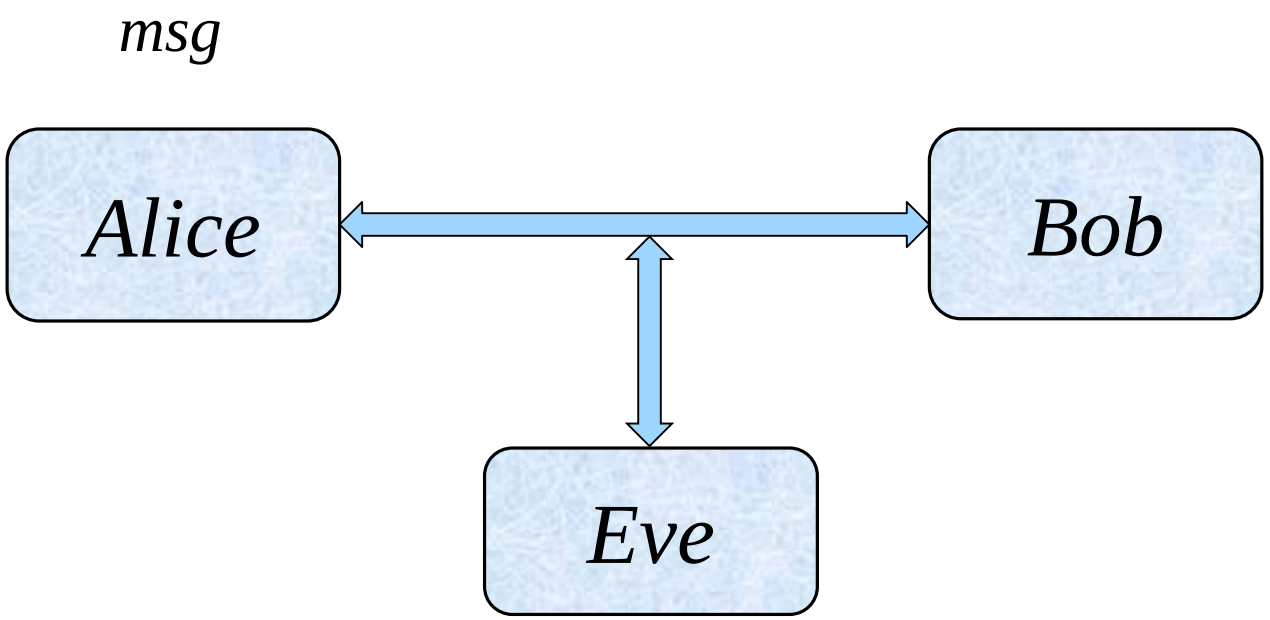
Applicazioni della Crittografia

- Memorizzazione sicura di dati, anche a prova di servizi segreti
- Trasmissione sicura di dati (soprattutto se sensibili)
  - es: carta di credito, per acquisti su siti di e-commerce: SSL/TLS
  - telefonia mobile
- DRM per la protezione di e-book, audio, video, software
- Firme digitali, anche di gruppo
- Crittografia omomorfica e Cloud Computing
- Critto-monete e smart contracts: Bitcoin, Litecoin, Ether(eum)…
- Condivisione di segreti
- Dimostrazioni interattive e zero-knowledge

Schema di comunicazione

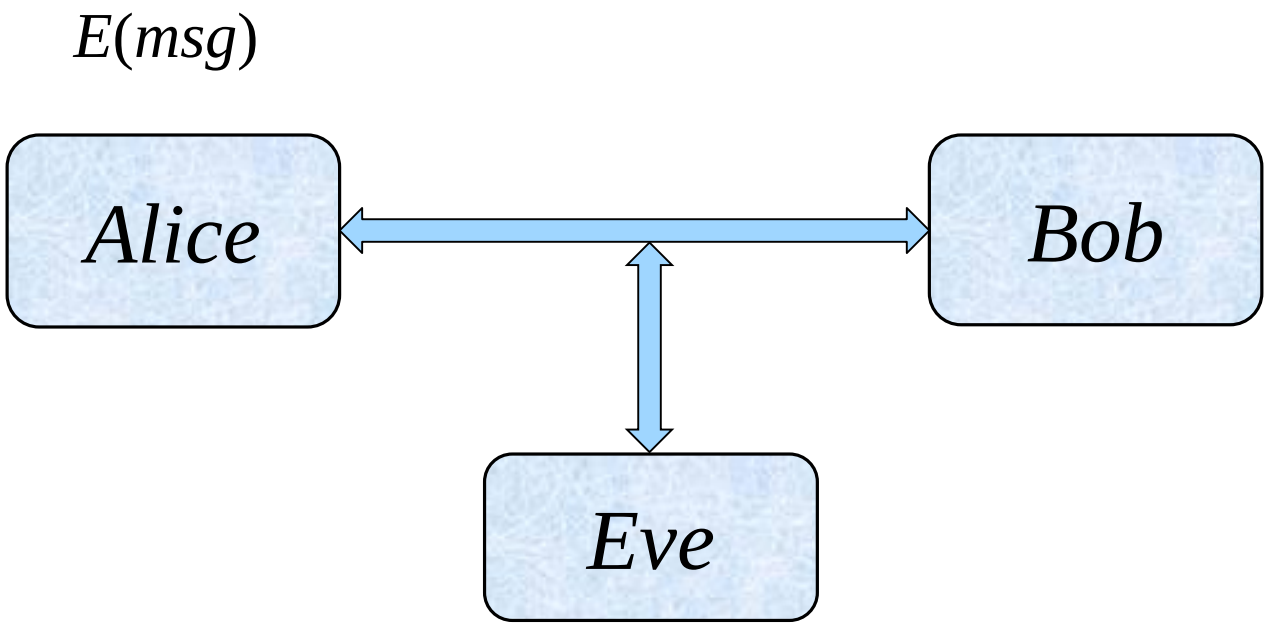


Trasmissione di un messaggio



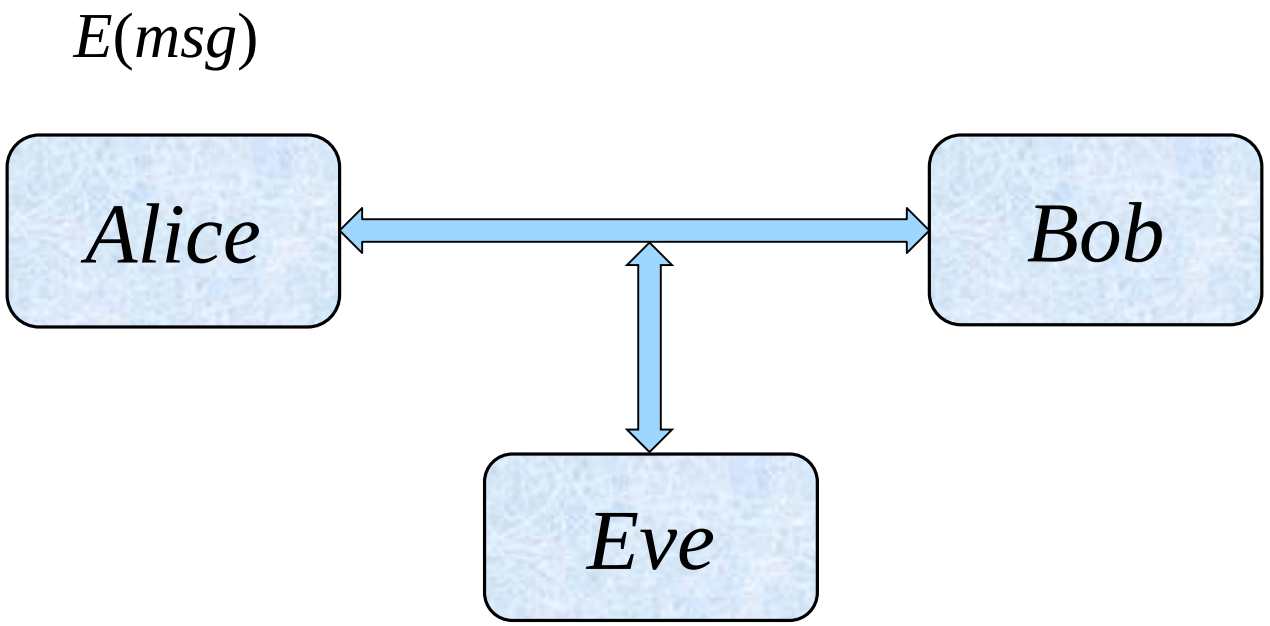
- Se *Alice* spedisce il messaggio così com'è, *Eve* lo può intercettare !

Trasmissione di un messaggio



- Idea: *Alice* codifica il messaggio, così se *Eve* lo intercetta non ci capisce niente !

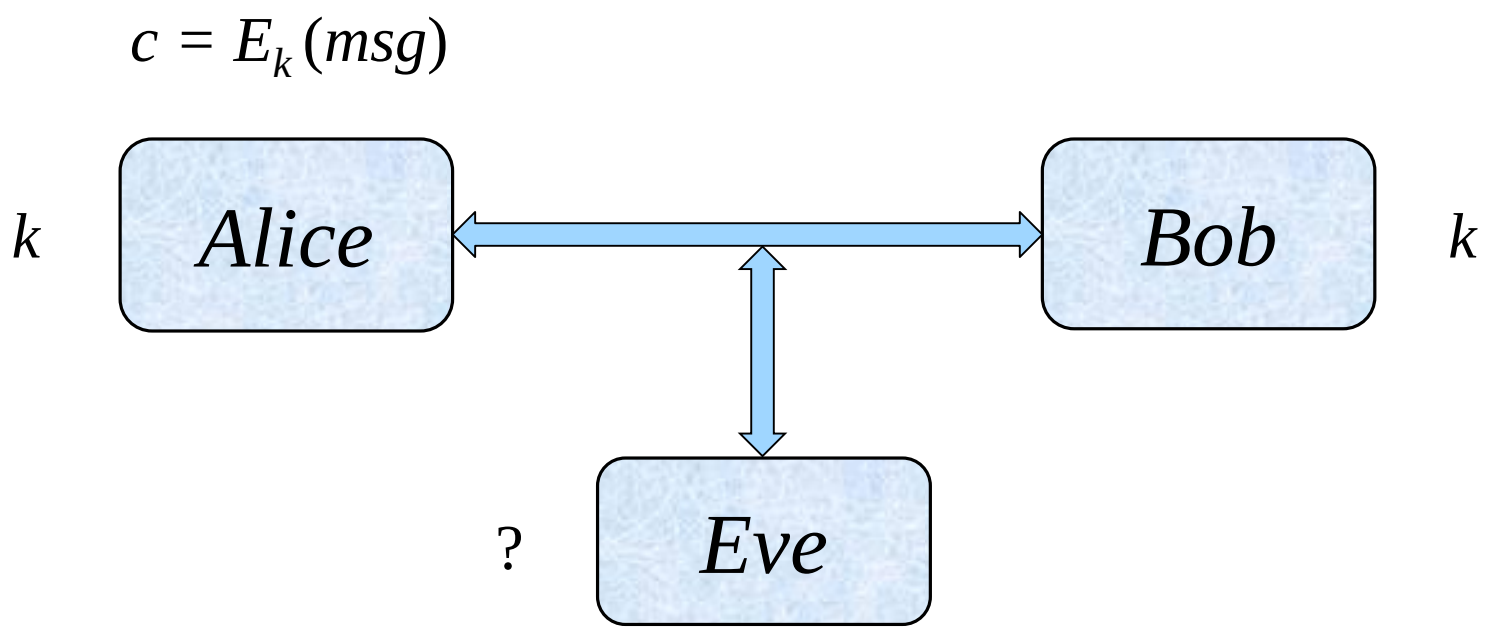
Trasmissione di un messaggio



- Problema: ma anche *Bob* non ci capisce niente !

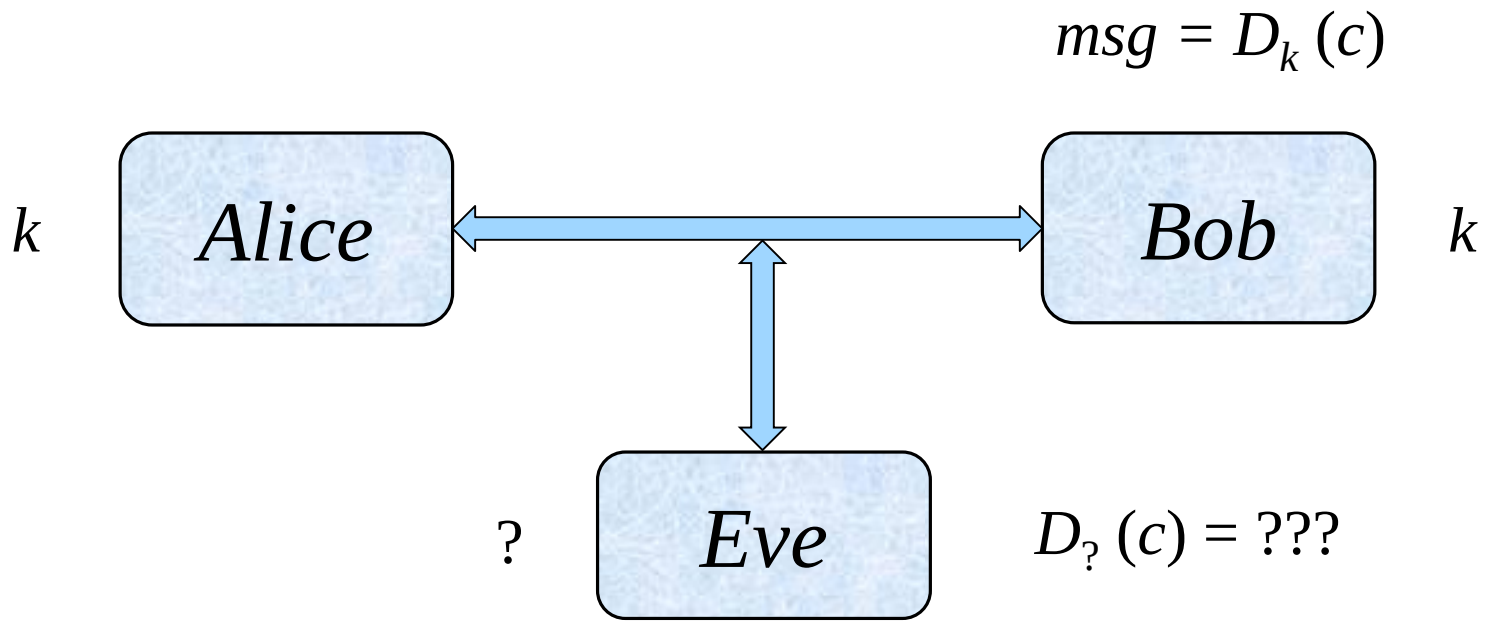
  ⇨ Bob deve sapere come decodificare il messaggio

Trasmissione di un messaggio



- Idea: la codifica dipende da un'informazione segreta (detta chiave), conosciuta solo da *Alice* e *Bob*

Trasmissione di un messaggio



- Così facendo, solo *Bob* riesce a decifrare correttamente

Obiettivi della Crittografia

- Gli attacchi possibili dipendono da cosa può fare *Eve* sul canale (leggere e/o scrivere), e (solitamente) dalla sua potenza di calcolo

Osservazione: non è detto che *Alice* e *Bob* siano i buoni, e *Eve* il cattivo!

Quindi, la Crittografia ha due obiettivi, in contrapposizione fra loro:

- studiare e implementare crittosistemi sicuri (Crittografia)
- analizzare i crittosistemi esistenti, al fine di scoprirne eventuali vulnerabilità (Crittoanalisi)

Su alcuni libri di testo vecchi, si parla di:

Crittografia + Crittoanalisi = Crittologia

Alcuni requisiti…

- $\forall m \; \forall k \;\; D_k(E_k(m)) = m$
- $\forall m \; \forall k$, il calcolo di $c = E_k(m)$ deve essere facile
- Deve essere estremamente difficile scoprire m dalla sola conoscenza di $c = E_k(m)$
- Deve essere invece facile calcolare m dalla conoscenza di c e di k
- Deve essere estremamente difficile scoprire k dalla conoscenza di $c = E_k(m)$ e di m

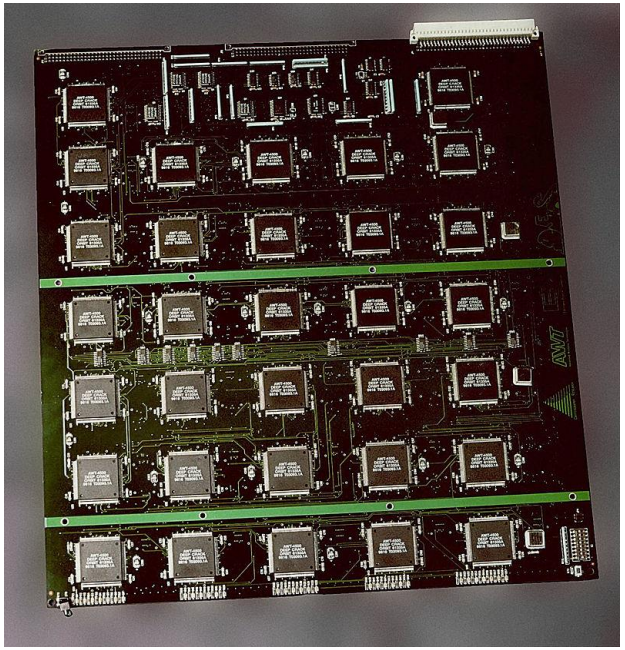
---

- Principio di Kerckhoffs (1880 ca.):

  In un sistema crittografico, la segretezza non deve risiedere nelle funzioni E e D, bensì in una piccola informazione detta chiave

Crittosistemi simmetrici attuali

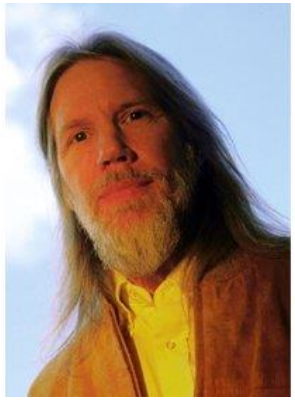
- DES (o meglio, DEA):
  - sviluppato da IBM nel 1977
  - primo crittosistema standard: US FIPS 46
  - chiave segreta da 56 bit
  - cifra blocchi da 64 bit
  - considerato non più sicuro
- 3DES: due chiavi da 56 bit
- AES (Rijndael)
  - inventato da J. Daemen e V. Rijmen
  - standard attuale: US FIPS 197 (2002)
  - chiave segreta da 128, 192 o 256 bit
  - cifra blocchi da 128 bit

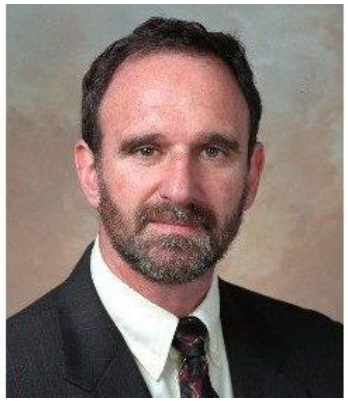


EFF DES Cracker (1998)
1800 processori, 250.000 $
Trova la chiave segreta in circa due giorni

Crittografia a chiave pubblica

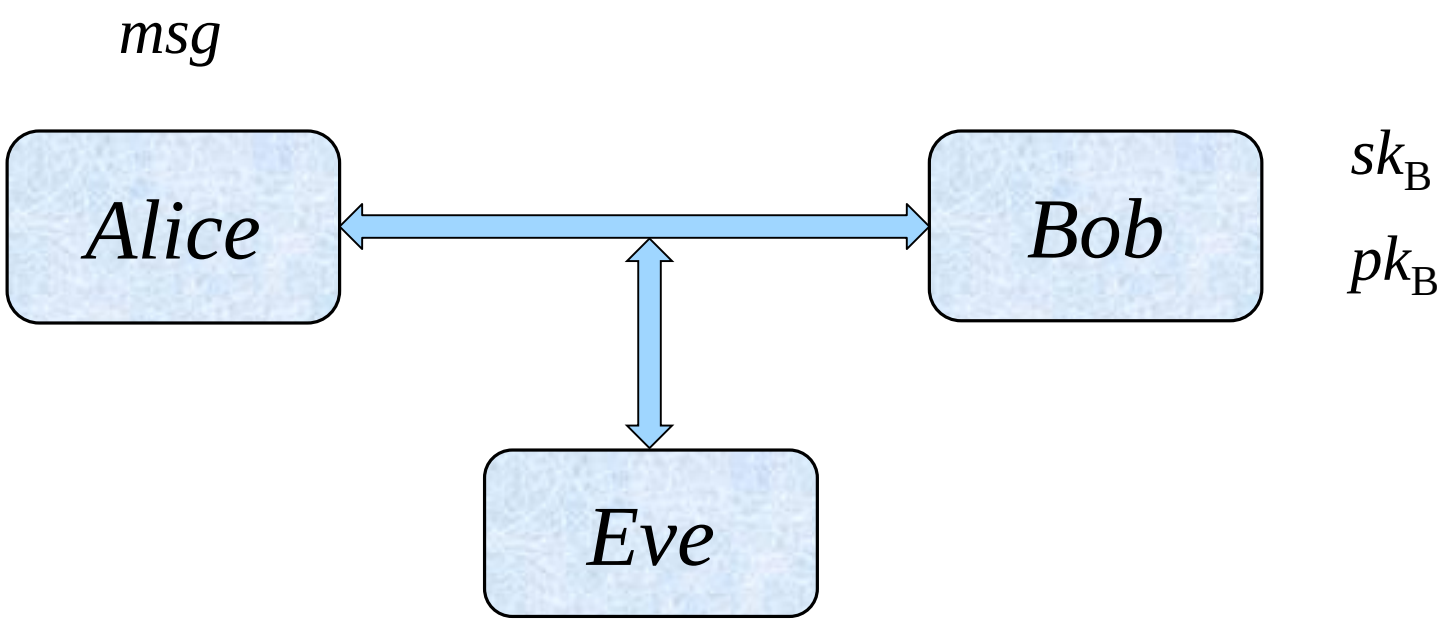
- È possibile far comunicare *Alice* e *Bob* in maniera sicura su un canale pubblico, anche se non si sono mai scambiati una chiave segreta?
- Idea: le chiavi usate per cifrare e per decifrare sono diverse
  - la chiave per decifrare "non può" essere ricavata da quella per cifrare
  - idea realizzata negli anni 1976 – 1977
- Ognuno crea una propria coppia di chiavi
  - la chiave per cifrare viene resa pubblica
  - la chiave per decifrare rimane segreta
- Quindi:
  - chiunque è in grado di cifrare
  - solo il destinatario è in grado di decifrare





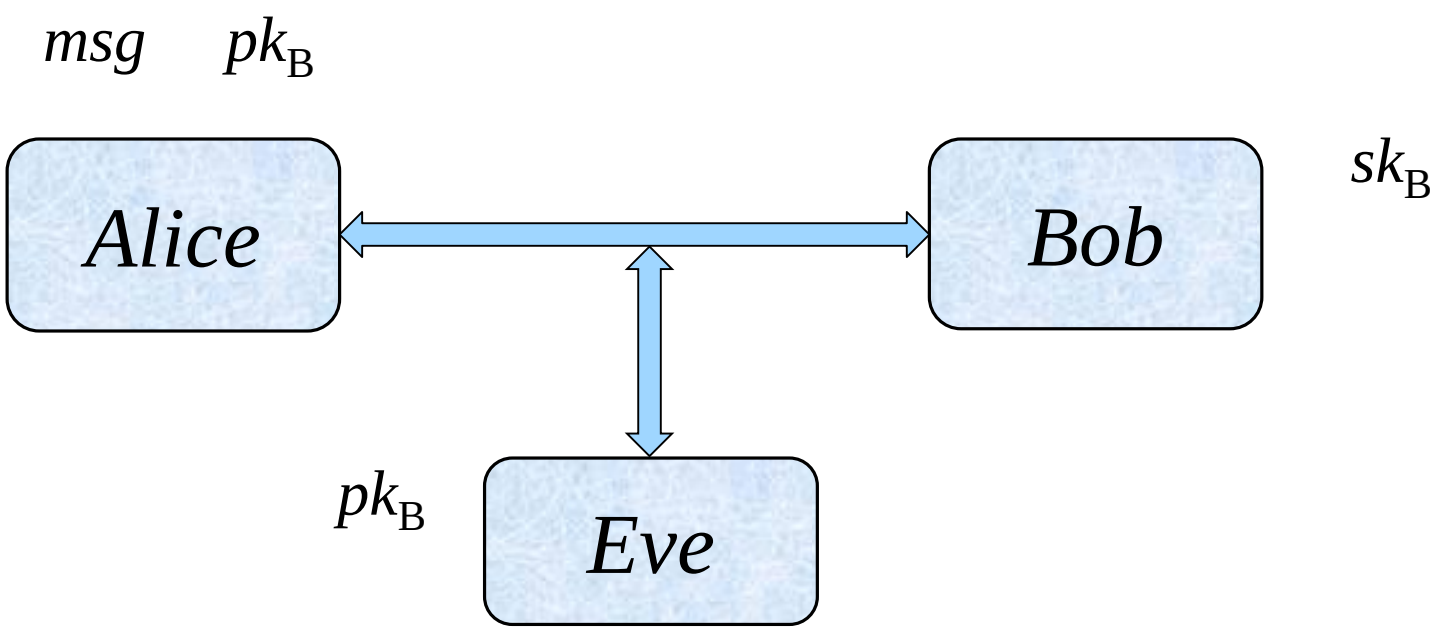
Whitfield Diffie (1944 –)
Martin Hellman (1945 –)

Trasmissione di un messaggio



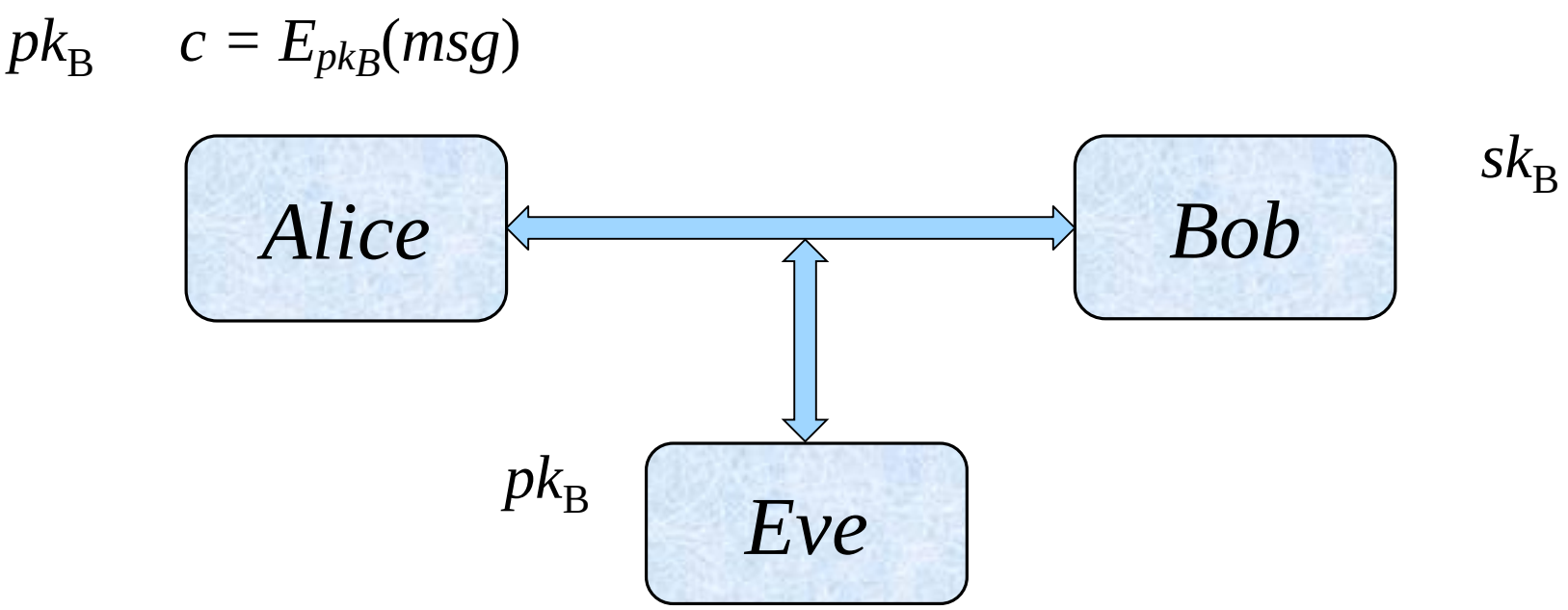
- *Alice* vuole inviare un messaggio a *Bob*

Trasmissione di un messaggio



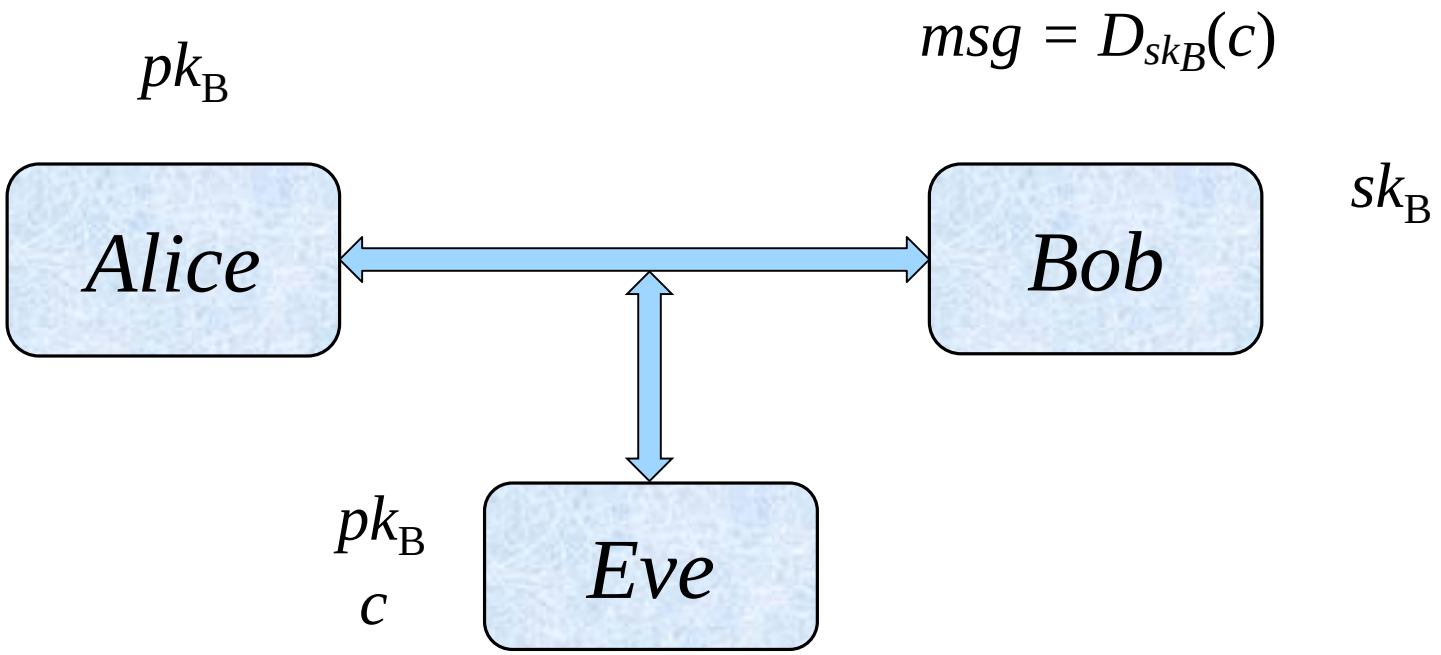
- *Alice* chiede a *Bob* la sua chiave pubblica pk_B (e *Eve* la intercetta)

Trasmissione di un messaggio



- *Alice* costruisce un testo cifrato che solo *Bob* può decifrare, e glielo invia (Eve lo intercetta)

Trasmissione di un messaggio



- *Bob*, usando la propria chiave segreta sk_B, decifra e riottiene il messaggio

Alcuni requisiti…

- $\forall m \;\; D_{sk}(E_{pk}(m)) = m$
- $\forall m \; \forall pk$, il calcolo di $c = E_{pk}(m)$ deve essere facile
- Deve essere estremamente difficile scoprire m dalla sola conoscenza di $c = E_{pk}(m)$
- Deve essere invece facile calcolare m dalla conoscenza di c e di sk
- Deve essere estremamente difficile scoprire sk dalla conoscenza di $c = E_{pk}(m)$, di m e di pk

---

- Osservazione: le cose sono in realtà un po' più complicate...

  Così com'è, il canale è cifrato ma non autenticato: quando *Alice* riceve la chiave pubblica di *Bob*, come fa a essere sicuro che non sia invece quella di *Eve*?

Cosa vuol dire facile o difficile?

- Sono nozioni di Complessità Computazionale:
  - facile: esiste un algoritmo polinomiale eseguibile da una Macchina di Turing deterministica
  - difficile: non esiste (o non si sa se esiste) tale algoritmo
- Alcuni problemi, infatti, sono difficili da risolvere
- Cosa ancora più interessante: le funzioni one-way
  - facili da calcolare: dato x, calcolare $f(x)$
  - difficili da invertire: dato $y = f(x)$, trovare x
- Cosa ancora più interessante: funzioni one-way con trapdoor
  - facili da calcolare
  - difficili da invertire, se non si conosce la trapdoor
  - facili da invertire, se si conosce la trapdoor

Esempio di funzione one-way

- Un esempio di funzione che sembrerebbe essere one-way è il prodotto tra due numeri primi:
  - dati due numeri primi p e q, è facile calcolare $n = pq$
  - dato n, è difficile ricavare p (o q)

Fattorizzazione

- Il tempo di calcolo si conta in funzione del numero di bit di n
  - posto $m = \log_2 n$, provare a dividere per tutti i numeri compresi tra 2 e $\sqrt{n}$ richiede tempo esponenziale:

$$O(\sqrt{n}) = O(\sqrt{2^m}) = O(2^{m/2})$$

  - tuttora, non si sa se esiste un algoritmo polinomiale !

Fattorizzazione

- Esempio di istanza di Fattorizzazione:

RSA-768 = 123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985 6902143413

Fattorizzazione

- Esempio di istanza di Fattorizzazione:

RSA-768 = 123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985 6902143413

= 33478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489

× 36746043666799590428244633799627952632291581643430876426760322838157396665112792333734171433968102700927987363089 17

Logaritmi discreti

- Dato un numero primo p, e un generatore g di $\mathbb{Z}_p$
  (significa che $\mathbb{Z}_p = \{g^0, g^1, g^2, \ldots, g^{p-2}\}$), definiamo la funzione:
  $f(x) = y = g^x \bmod p$ (esponenziazione modulare)
- Problema del logaritmo discreto:
  dato y, trovare x tale che $y = g^x \bmod p$, cioè $x = \log_g y \bmod p$
- Non si conosce nessun algoritmo efficiente (polinomiale)!

- Provate a calcolare a mano $\log_2 6113 \bmod 25307$ (= 3578)
- Poi, considerate che in realtà x, y e p dovrebbero essere numeri di almeno 1024 bit = 128 cifre

Firme digitali

- Consentono di autenticare chi ha generato un messaggio
  - solo chi possiede una informazione segreta (chiave) può averlo creato
- La firma cambia a seconda del messaggio firmato
- Chiunque può verificare la firma di un messaggio

- Ogni crittosistema a chiave pubblica può essere usato per firmare e verificare firme
- Algoritmi (standard) di firma digitale:
  - DSA: FIPS-186 (1991), FIPS-186-4 (2013)
  - ECDSA (1991), basato su curve ellittiche, standard ISO 14888 (1998), ANSI X9.62 (1999), IEEE P1363 2 (2000)
    - usato per firmare le transazioni in Bitcoin

Funzioni di hash

Mappano messaggi arbitrariamente lunghi in messaggi di lunghezza prefissata, in modo tale che valgano le seguenti proprietà:

- *first pre-image resistance*: data la funzione $h: X \rightarrow Y$, e un valore $y \in Y$, è difficile trovare un valore $x \in X$ tale che $h(x) = y$
- *second pre-image resistance*: data la funzione $h: X \rightarrow Y$, e un valore $x \in X$, è difficile trovare un valore $x' \in X$ tale che $x \neq x'$ e $h(x) = h(x')$
- *collision resistance*: data la funzione $h: X \rightarrow Y$, è difficile trovare due valori $x, x' \in X$ tali che $x \neq x'$ e $h(x) = h(x')$
- Data la funzione $h: X \rightarrow Y$, e un valore $x \in X$, è facile calcolare $h(x)$

Le funzioni hash calcolano un'**impronta** del messaggio:

$h(x) =$ 

SHA-256("Lezione Lincea") =

3344867ddcc5bd2e0f3dac64bc99ca67df46ae71a09e5c55f3ad898e82048194

Torniamo a Bitcoin

- I membri della rete P2P si chiamano miners (minatori)
  - « scavano » bitcoin come se fosse oro in una miniera
  - la quantità di oro è limitata → max 21 mln di bitcoin
  - vengono premiati quando riescono ad aggiungere un nodo alla blockchain
  - scavare all'inizio è facile, poi via via più difficile → proof-of-work via via più difficile: collisione di una funzione di hash su insiemi via via più piccoli di valori
  - si aggiunge un nodo alla blockchain ogni circa 10 min

(Immagine tratta da: http://people.fas.harvard.edu/~maryphotiades/bitcoinmining.html

Mining di bitcoin

https://news4c.com/bitcoin-mining-and-everything-you-need-to-know-about-it/

Mining casalingo…

… e mining professionale

http://www.livebitcoinnews.com/socially-optimal-bitcoin-mining-pools/

Come avviene una transazione

- Supponiamo che Bob voglia inviare 1 BTC ad Alice
- Usando il proprio client / wallet, inserisce la cifra e l'indirizzo (chiave pubblica) di Alice
- Eventualmente aggiunge una fee per il miner
- Firma (con ECDSA) la transazione con la propria chiave segreta
- Invia la richiesta di transazione alla rete P2P

- Alcuni miner selezionano la transazione e la aggiungono a un blocco
- Aggiungono al blocco l'hash (SHA-256) dell'ultimo blocco della blockchain
- Eseguono il proof-of-work (collisione hash di SHA-256)
- Il primo che termina il proof-of-work aggiunge il nodo alla blockchain

Alcuni dettagli

- I miner selezionano solo transazioni valide
- Ogni blocco contiene molte transazioni (un migliaio)
- Se Bob ha 10 BTC, per dare 1 BTC deve darsi un resto di 9 BTC
  - il totale in input deve essere uguale al totale in output
- Bob può pagare contemporaneamente più persone

- I miner selezionano prima le transazioni con fee più alte
- Il miner che completa la proof-of-work sottopone il risultato alla rete, che la valida
- Se il blocco non viene validato, diventa orfano
- Man mano che Alice riceve validazioni, è sempre più portata a credere che la transazione sia andata a buon fine

Alcuni problemi

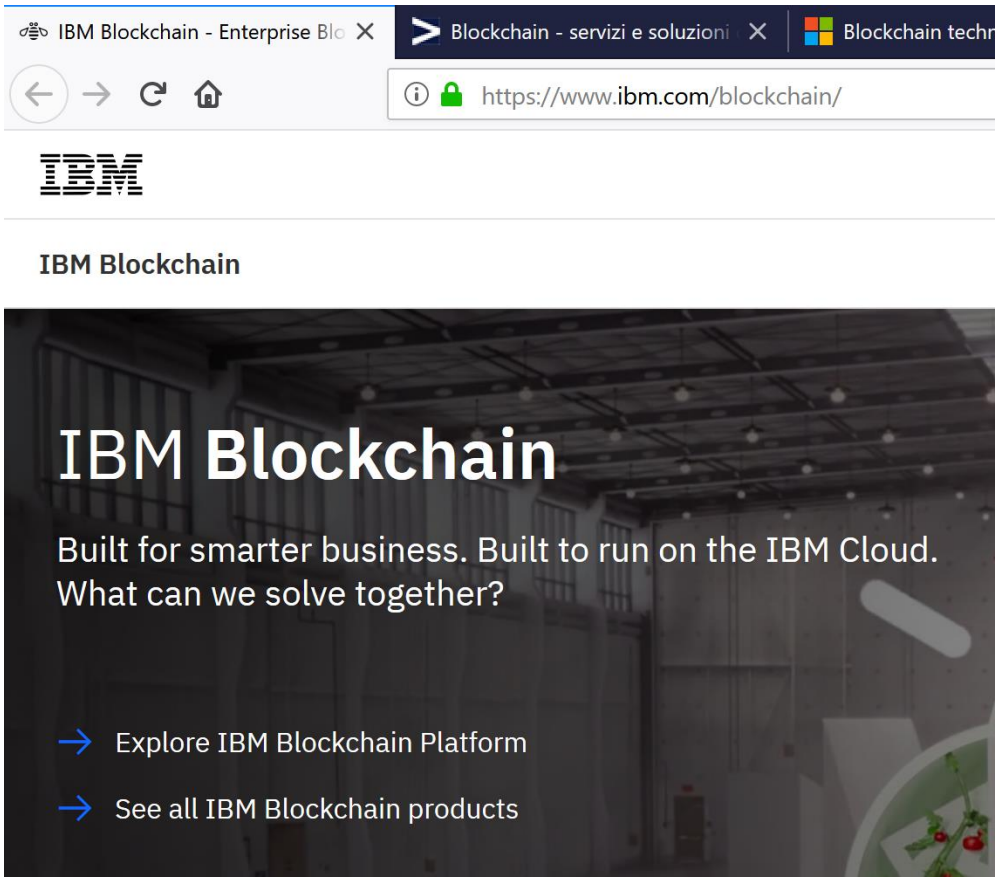
- Molte transazioni con fee = 0 sono zombie
- I miner sono tutti contro tutti, e fanno un lavoro duplicato
- Il consumo di corrente è altissimo
- Fare il miner è diventata prima una professione, poi un investimento
- Attacco del 51%

- Il numero di transazioni globale è di un migliaio ogni 10 min
  - fork: Bitcoin Cash (1 agosto 2017)
- La validazione di una transazione può richiedere anche un'ora
- Lo pseudo-anonimato ha portato a usare Bitcoin per il riciclaggio di denaro sporco

Il futuro delle blockchain

- Dipende dalla (vostra) fantasia!
- Un sacco di applicazioni possibili
  - registri di beni o di dati verificati (medici, agro-alimentari, …)
  - colored coins
  - smart contracts, calcolo distribuito
  - prodotti finanziari: future, opzioni, derivati, …
- Un sacco di varianti possibili
  - proprietarie (permissioned) o distribuite (permissionless)
  - side-chains
- Diverse implementazioni
  - lightning networks

Le grosse aziende si sono mosse…



Envisioning a fully transparent food system

With IBM Blockchain, retailers and suppliers can trace a food item from farm to store in seconds.

→ See more on supply chain

Trusted identity with SecureKey

SecureKey builds identity attribute exchange networks with major Canadian banks for triple-blind data privacy.

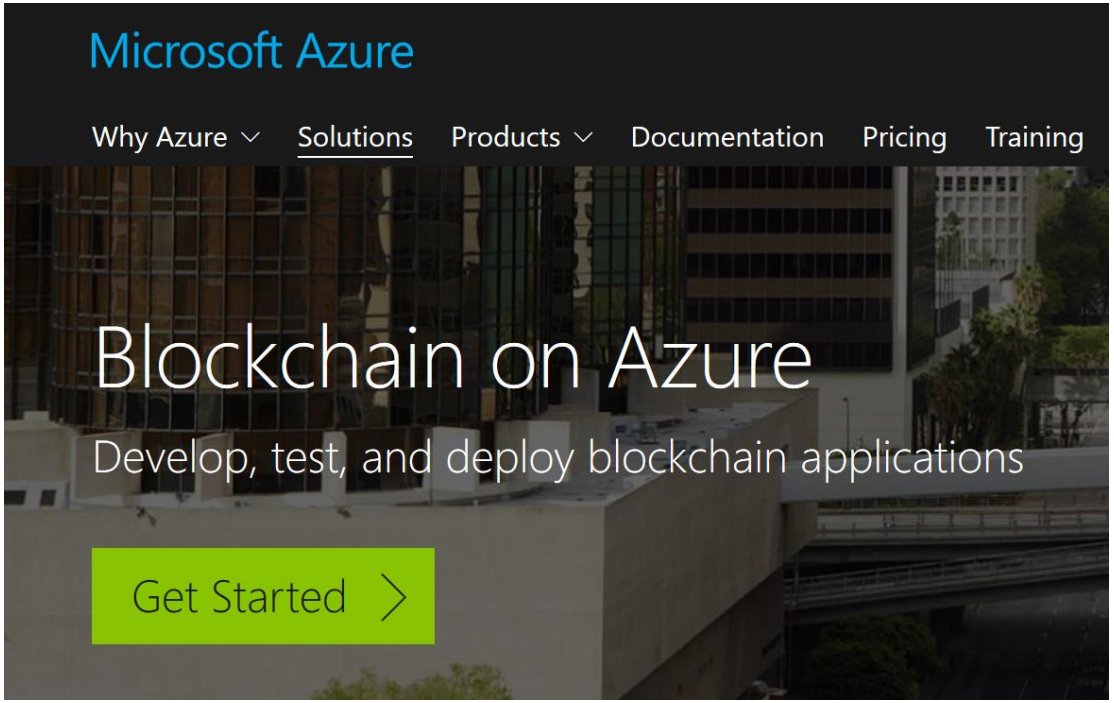
→ See more on identity

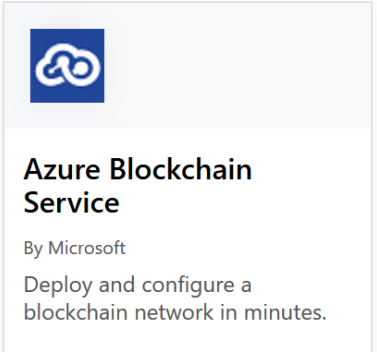
Blazing a path for blockchain in private equity

Northern Trust manages the administration of private equity funds on blockchain to bring trust and efficiency.

→ See more on financial services

Le grosse aziende si sono mosse…





Blockchain >

- Single node ledger
- Multi-node ledger
- Tools
- App accelerators

Ethereum

- Fondata nel 2014 da Vitalin Buterik (e altri)
- Piattaforma per implementare applicazioni decentralizzate
- Ha una sua crittomoneta: Ether (ETH)
- Fornisce allo sviluppatore una rete di macchine virtuali
- Programmabile in linguaggio ad alto livello: Solidity
- Consente di implementare smart contracts

ethereum
BLOCKCHAIN APP PLATFORM

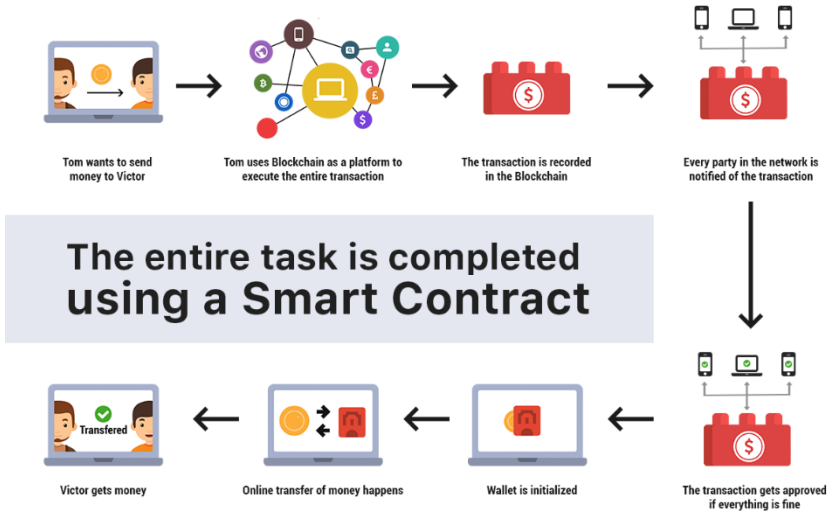
Solidity

What is CryptoKitties?

CryptoKitties is a game centered around breedable, collectible, and oh-so-adorable creatures we call CryptoKitties! Each cat is one-of-a-kind and 100% owned by you; it cannot be replicated, taken away, or destroyed.

Smart contracts

- Contratti tra più parti (come quelli legali) codificati in un linguaggio di programmazione
- Le parti, di solito, non si fidano l'una dell'altra
- Una volta avviati, la loro esecuzione è automatica
- Possono essere collegati a oggetti fisici:
  - Pagamenti a rate di beni
  - Dispositivi IoT
    (vedi IOTA: https://iota.org/)
- Problemi:
  - Interpretazione legale
  - Sono « disumani »
- **Se ne parlerà sempre di più!**



https://www.openxcell.com/smart-contracts-audit

Grazie per l'attenzione !

BiS Lab

bislab@unimib.it

Claudio Ferretti

Alberto Leporati

alberto.leporati@unimib.it

Ci vediamo alla Digital Week

Sabato 17 marzo

Andrea Rossetti