Kryptografiska mekanismer och valutor

Daniel Bosk[1]

Avdelningen för informations- och kommunikationssytem (IKS),
Mittuniversitetet, Sundsvall.

hashsign.tex 1020 2013-05-06 12:27:13Z danbos

Mittuniversitetet
MID SWEDEN UNIVERSITY

Översikt

1. Digitala signaturer
   - Asymmetrisk kryptering
2. Hashfunktioner
   - Introduktion till hashfunktioner
   - Formell behandling av hashfunktioner
   - Message Authentication Code (MAC)
3. Kryptografiska valutor – Bitcoin
   - Idé
   - Överföringar
   - Decentralisering
   - Att dela upp mynt
   - Personlig integritet

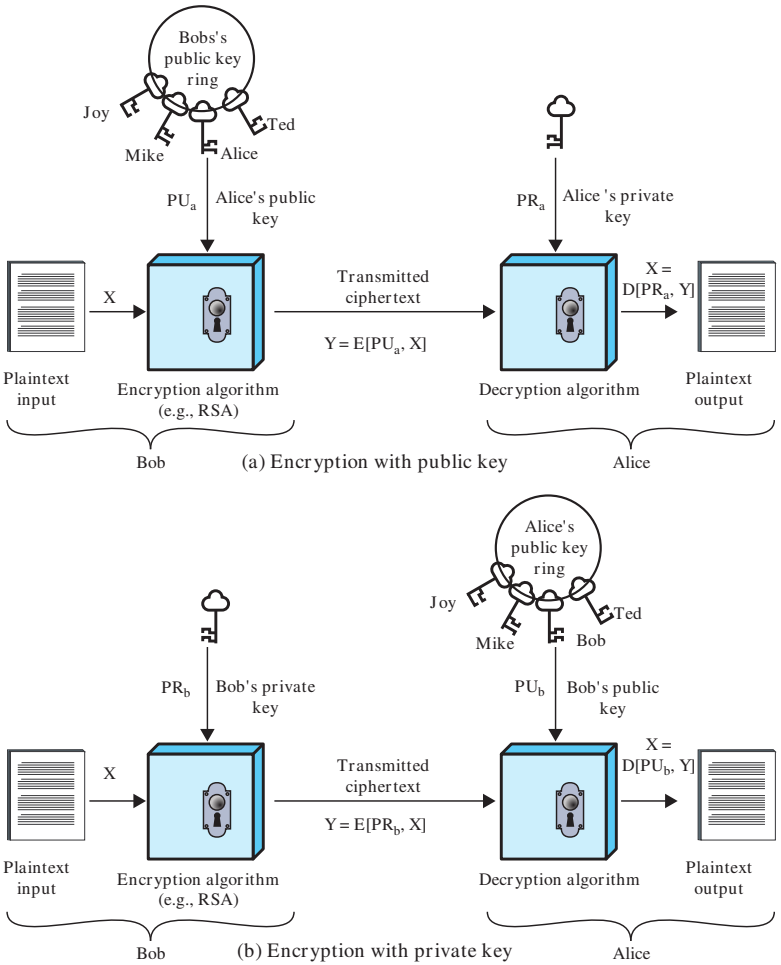
Litteratur

Föreläsningen går igenom resterande delar av kapitel 5 *Cryptography* i [And08] och hur kryptovalutan Bitcoin fungerar och vilka grunder den bygger på [Pec12; Kam13; Nak08; BBSU12].

Översikt

Översikt

Asymmetrisk kryptering



Figur : Översikt av asymmetrisk kryptering. Bild: [Sta11].

Asymmetrisk kryptering

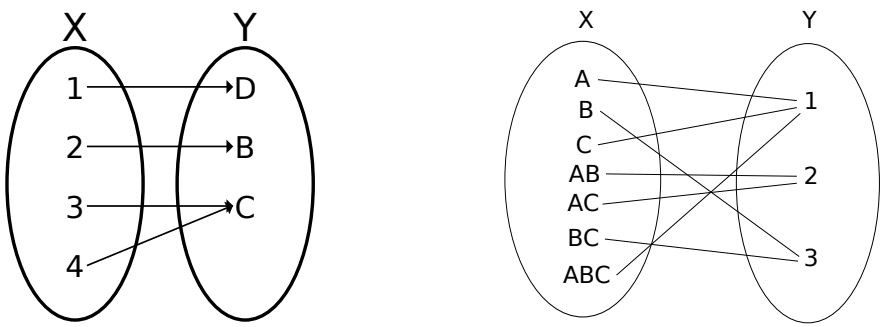
- $E_{k_A}(m) = c \longrightarrow D_{k_A}(c) = E_{k_A^{-1}}(c) = m$
- $D_{k_A}(m) = E_{k_A^{-1}}(m) = c \longrightarrow E_{k_A}(c) = m$

Översikt

Översikt

Introduktion till hashfunktioner

- En hashfunktion är en funktion $h\colon X \to Y$, där X är en möjligen oändlig mängd och Y är en ändlig mängd.
- Den är således en icke-injektiv surjektiv funktion och saknar invers $h^{-1}\colon Y \to X$ sådan att $h^{-1}(h(x)) = x$ för alla $x \in X$.



(a) $h\colon X \to Y$

(b) $h'\colon X \to Y$

Figur : Två icke-injektiva surjektiva funktioner h respektive h'.

Introduktion till hashfunktioner

- Finns många olika hashfunktioner:
  - MD5,
  - SHA1,
  - SHA256,
  - SHA512.
- Tillämpningsområdet är stort:
  - verifiera integritet hos filer,
  - snabb sökning i datastrukturer,
  - digitala signaturer,
  - skydda lösenord.

Översikt

Formell behandling av hashfunktioner

Definition

En *hashfamilj* är en tupel $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$, där

- $\mathcal{X}$ är mängden av möjliga *meddelanden*.
- $\mathcal{Y}$ är en ändlig mängd av möjliga *meddelandesammandrag*.
- $\mathcal{K}$ är en ändlig mängd av möjliga nycklar.
- För varje nyckel $k \in \mathcal{K}$ finns en hashfunktion $h_k \in \mathcal{H}$ sådan att $h_k \colon \mathcal{X} \to \mathcal{Y}$.

Mittuniversitetet
MID SWEDEN UNIVERSITY

Formell behandling av hashfunktioner

- $\mathcal{X}$ kan vara ändlig eller oändlig, men alltid $|\mathcal{X}| \geq |\mathcal{Y}|$.
- Vissa hashfunktioner saknar nycklar, då är $|\mathcal{K}| = 1$.
- Låt $\mathcal{Y}^{\mathcal{X}}$ beteckna mängden av alla funktioner från $\mathcal{X}$ till $\mathcal{Y}$, då är $|\mathcal{Y}^{\mathcal{X}}| = |\mathcal{Y}|^{|\mathcal{X}|}$.

Formell behandling av hashfunktioner

Preimage resistant eller *one-way*

Inversa bilden (*preimage*)

1. Given hashfunktionen $h\colon\ \mathcal{X} \to \mathcal{Y}$ och element $y \in \mathcal{Y}$.
2. Hitta $x \in \mathcal{X}$ sådant att $h(x) = y$.

Formell behandling av hashfunktioner

Second preimage resistant

Andra inversa förbilden (*second preimage*)

1. Given hashfunktionen $h\colon\ \mathcal{X} \to \mathcal{Y}$ och element $x \in \mathcal{X}$.
2. Hitta $x' \in \mathcal{X}$ sådant att $x' \neq x$ och $h(x') = h(x)$.

Formell behandling av hashfunktioner

Collision resistant

Kollision

1. Given hashfunktionen $h\colon\ \mathcal{X} \to \mathcal{Y}$.
2. Hitta $x, x' \in \mathcal{X}$ sådana att $x' \neq x$ och $h(x') = h(x)$.

Formell behandling av hashfunktioner

Random Oracle Model

- Idealisering av en hashfunktion.
- Kan liknas vid ett orakel som ger slumpmässiga svar på frågor.
- Men vid upprepningar ska samma svar ges.
- En funktion $h \in \mathcal{Y}^{\mathcal{X}}$ väljs slumpmässigt, vi får enbart ställa frågor som "vad är $h(x)$?"
- Innan vi ställer frågan $h(x)$ vet vi ingenting om h.
- Efter att vi ställt frågan $h(x)$ och erhållit svaret y, då vet vi enbart att $h(x) = y$.

Formell behandling av hashfunktioner

- Det går att visa att om man kan hitta en andra invers avbildning, då kan man hitta en kollision.
- Det går även att visa att om man kan hitta en invers avbildning, då kan man hitta en kollision.
- Följaktligen, om en hashfunktion är *collision resistant*, då är den även *preimage* och *second preimage resistant*.

Formell behandling av hashfunktioner

Sats (Oberoendesatsen)

Antag att $h \in \mathcal{Y}^{\mathcal{X}}$ *väljs slumpmässigt. Låt* $\mathcal{X}_0 \subseteq \mathcal{X}$. *Antag att värdet* $h(x)$ *bestämts genom att fråga oraklet om och endast om* $x \in \mathcal{X}_0$. *Då gäller att*

$$\Pr(h(x) = y) = \frac{1}{|\mathcal{Y}|}$$

för alla $x \in \mathcal{X} \setminus \mathcal{X}_0$ *och alla* $y \in \mathcal{Y}$.

Formell behandling av hashfunktioner

Algoritm (Hitta invers avbild)

input $h \in \mathcal{Y}^{\mathcal{X}}, y \in \mathcal{Y}, Q \in \mathbb{N}$
output x sådant att $h(x) = y$

Välj någon mängd $\mathcal{X}_0 \subseteq \mathcal{X}$ sådan att $|\mathcal{X}_0| = Q$.
for all $x \in \mathcal{X}_0$ **do**
  if $h(x) = y$ **then**
    return x
  end if
end for
return misslyckande

Formell behandling av hashfunktioner

Sats

För någon mängd $\mathcal{X}_0 \subseteq \mathcal{X}$ med $|\mathcal{X}_0| = Q$ är sannolikheten ϵ att algoritmen för att finna en inverterad avbildning lyckas

$$\epsilon = 1 - \left(1 - \frac{1}{|\mathcal{Y}|}\right)^Q.$$

Formell behandling av hashfunktioner

Bevis.

Fixera $y \in \mathcal{Y}$. Låt $\mathcal{X}_0 = \{x_1, \ldots, x_Q\}$ och låt E_i beteckna händelsen att $h(x_i) = y$. Det följer från oberoendesatsen att E_i är oberoende händelser och att $\Pr(E_i) = \frac{1}{|\mathcal{Y}|}$ för $1 \leq i \leq Q$. Då får vi

$$\Pr(E_1 \vee \cdots \vee E_Q) = 1 - \left(1 - \frac{1}{|\mathcal{Y}|}\right)^Q.$$

Då sannolikheten ϵ är oberoende av y och konstant, då måste sannolikheten vara densamma för alla $y \in \mathcal{Y}$ sammantaget. Q.E.D.

Formell behandling av hashfunktioner

Algoritm (Hitta kollision)

input $h \in \mathcal{Y}^{\mathcal{X}}, Q \in \mathbb{N}$
output $x, x' \in \mathcal{X}$ sådana att $x \neq x', h(x) = h(x')$

Välj någon mängd $\mathcal{X}_0 \subseteq \mathcal{X}$ sådan att $|\mathcal{X}_0| = Q$.
for all $x \in \mathcal{X}$ **do**
    Låt $y_x = h(x)$.
end for
if $y_x = y_{x'}$ för något $x \neq x'$ **then**
    return (x, x')
end if
return misslyckande

Formell behandling av hashfunktioner

Sats

För någon mängd $\mathcal{X}_0 \subseteq \mathcal{X}$ med $|\mathcal{X}_0| = Q$ är sannolikheten ϵ för att kollisionsalgoritmen lyckas följande:

$$\epsilon = 1 - \left(\frac{|\mathcal{Y}| - 1}{|Y|}\right)\left(\frac{|\mathcal{Y}| - 2}{|\mathcal{Y}|}\right)\cdots\left(\frac{|\mathcal{Y}| - Q + 1}{|\mathcal{Y}|}\right).$$

Formell behandling av hashfunktioner

Bevis.

Låt $\mathcal{X}_0 = \{x_1, \ldots, x_Q\}$. För $1 \leq i \leq Q$, låt E_i beteckna händelsen

$$h(x_i) \notin \{h(x_1), \ldots, h(x_{i-1})\}.$$

Det är klart att $\Pr(E_1) = 1$, då $h(x_1) \notin \emptyset$. Vi får genom induktion från oberoendesatsen att

$$\Pr(E_i \mid E_1 \wedge E_2 \wedge \cdots \wedge E_{i-1}) = 1 - \frac{i-1}{|\mathcal{Y}|} = \frac{|\mathcal{Y}| - i + 1}{|\mathcal{Y}|},$$

för $2 \leq i \leq Q$. Då har vi att

$$\Pr(E_1 \wedge \cdots \wedge E_Q) = \left(\frac{|\mathcal{Y}|}{|\mathcal{Y}|}\right)\left(\frac{|\mathcal{Y}| - 1}{|Y|}\right) \cdots \left(\frac{|\mathcal{Y}| - Q + 1}{|\mathcal{Y}|}\right).$$

Följaktligen blir sannolikheten för minst en kollision $1 - \Pr(E_1 \wedge \cdots \wedge E_Q)$. Q.E.D.

Formell behandling av hashfunktioner

- Vi hade att sannolikheten för ingen kollision är $\prod_{i=1}^{Q-1}(1 - 1/|\mathcal{Y}|)$.
- För små x gäller att $1 - x \approx e^{-x}$.
- Då får vi

$$\prod_{i=1}^{Q-1}(1 - 1/|\mathcal{Y}|) \approx \prod_{i=1}^{Q-1} e^{-i/|\mathcal{Y}|} = e^{\sum_{i=1}^{Q-1} i/|\mathcal{Y}|}.$$

- Följaktligen gäller $e^{\sum_{i=1}^{Q-1} i/|\mathcal{Y}|} \approx 1 - \epsilon$.
- Med lite omskrivningar får vi $Q \approx \sqrt{2|\mathcal{Y}| \log \frac{1}{1-\epsilon}}$.
- För $\epsilon = 1/2$ får vi då $Q \approx 1.17\sqrt{|\mathcal{Y}|}$.

Formell behandling av hashfunktioner

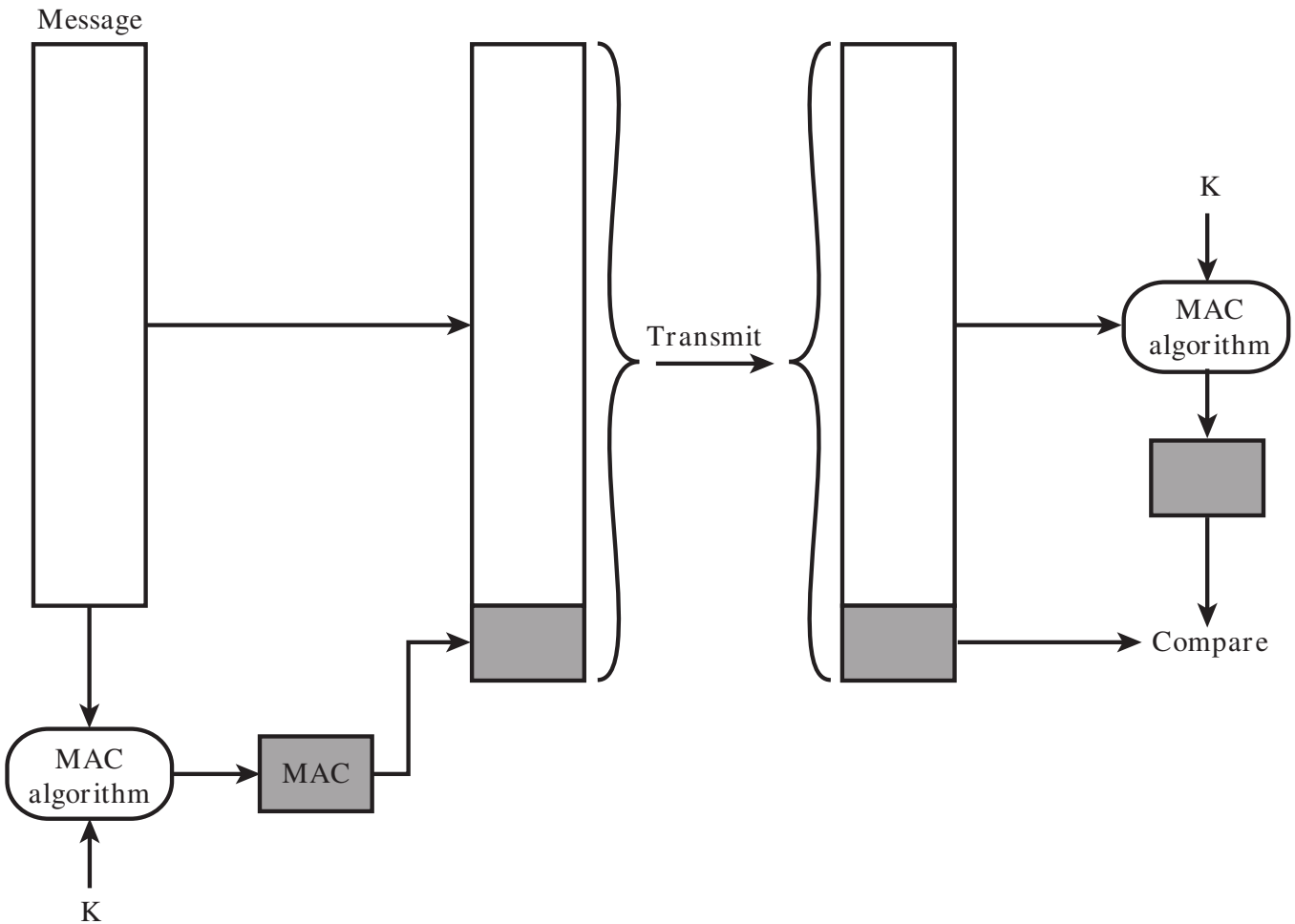
- Detta kallas födelsedagsparadoxen.
- Detta betyder att om $|\mathcal{Y}| = 365$, då är den 50 % sannolikhet att kollisionsalgoritmen finner en kollision då $Q = 23$.
- Om en fingeravtrycksläsare lagrar fingeravtryck som 20 bitar långa bitsträngar, då är det 50 % sannolikhet att två personer kan identifiera sig som varandra vid 1000 användare.
- Vi kan finna kollisioner med 50 % sannolikhet för en hashfunktion som har 256 bitars meddelandesammandrag med 2^{128} gissningar.

Formell behandling av hashfunktioner

MD5	Fullständigt knäckt; kan finna godtyckliga kollisioner, snabb att beräkna.
SHA1	Finns attacker som antyder att det går att finna kollisioner med $Q = 2^{69}$, borde vara $Q = 2^{80}$.
SHA256	Inga attacker som är märkbart lägre än $Q = 2^{128}$.
SHA512	Inga attacker som är märkbart lägre än $Q = 2^{256}$.

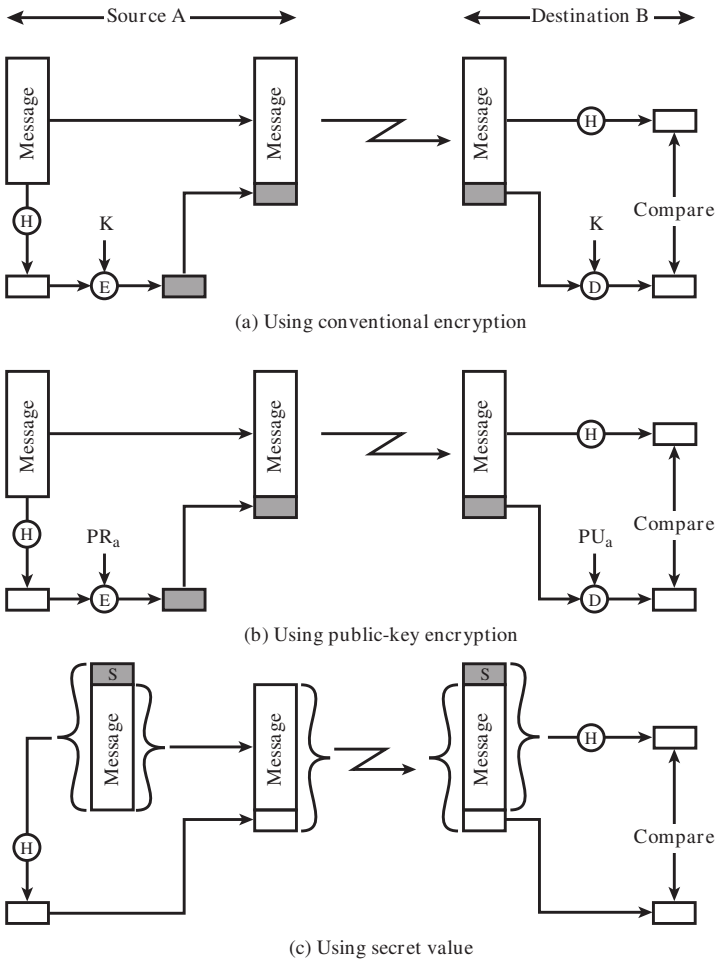
Översikt

Message Authentication Code (MAC)



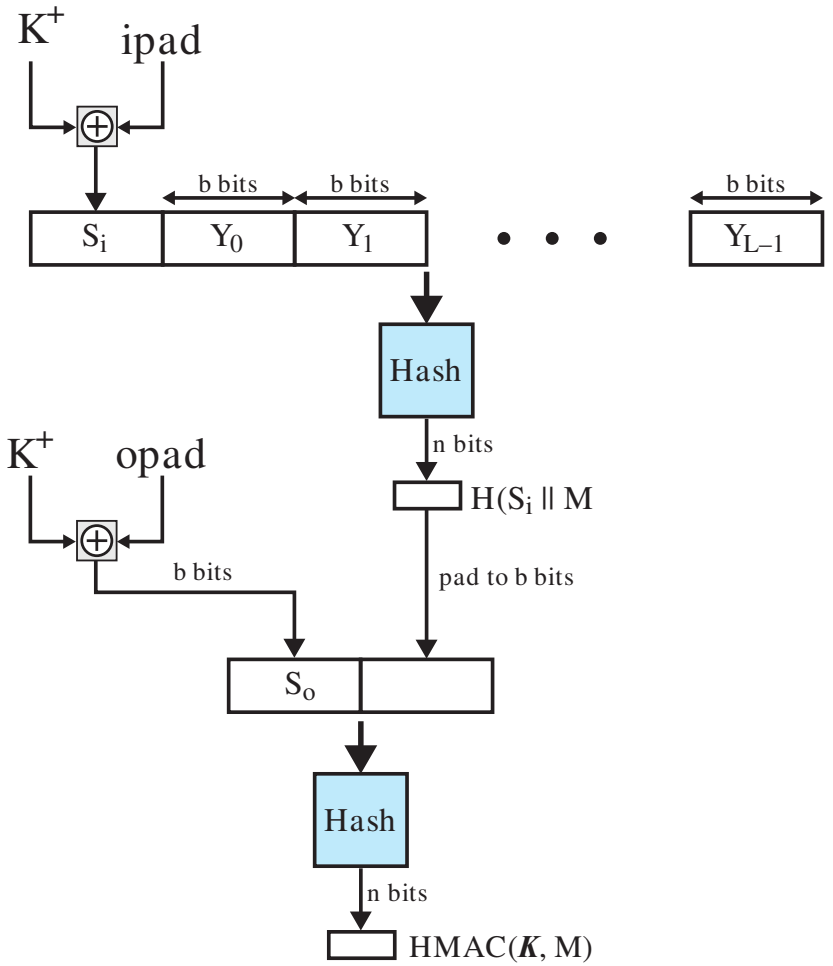
Figur : En översikt av en enkel MAC.

Message Authentication Code (MAC)



Figur : Exempel på olika former av MAC.

Message Authentication Code (MAC)



Figur : Hashbaserad MAC kallad HMAC,
$\mathrm{HMAC}(K, M) = h[(K^+ \oplus opad) \,||\, h[(K^+ \oplus ipad) \,||\, M]]$.

Översikt

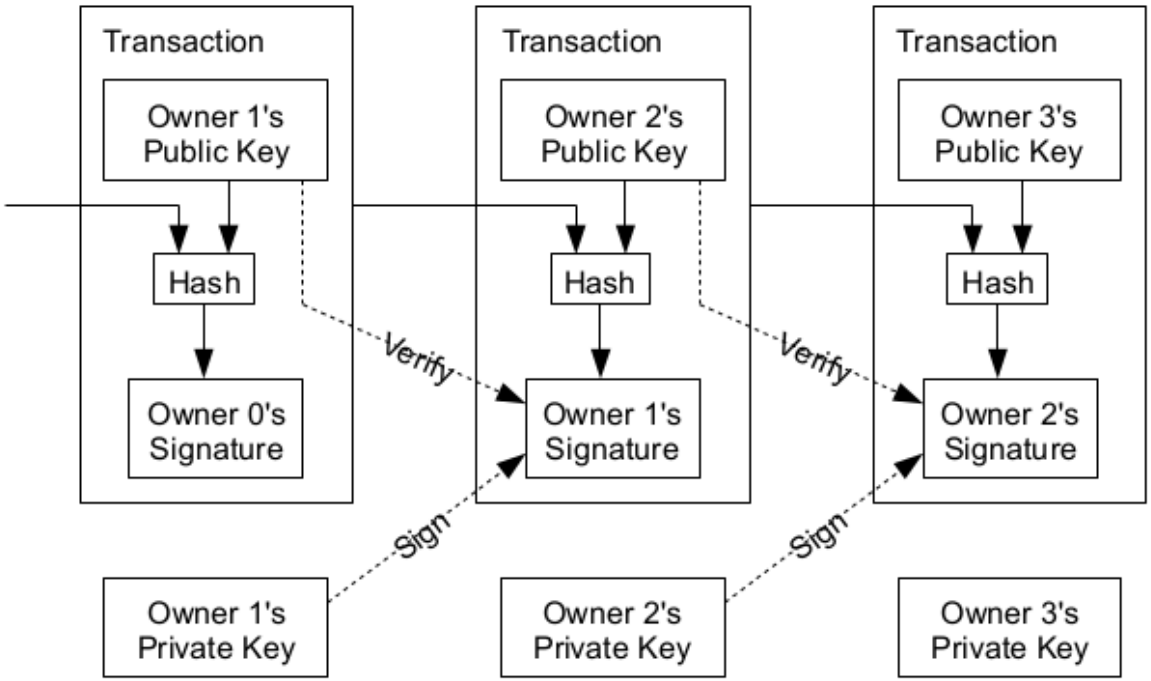
Översikt

Idé

- Decentraliserat, finns ingen central entitet för pålitlighet. Använder majoritetsomröstning.
- Idén är att alla överföringar är publika i en gemensam överföringshistorik.
- Säkerheten baseras på problem som inte enkelt kan lösas.
- *Bitcoin miners* löser dessa problem för att skapa bitcoins.
- Nya bitcoins skapas kontinuerligt med konstant avtagande hastighet.
- Ett bitcoins kan delas i godtyckligt antal delar, och delar kan sedan sättas ihop till större delar.
- Överföringar är permanenta.
- Låga överföringsavgifter.

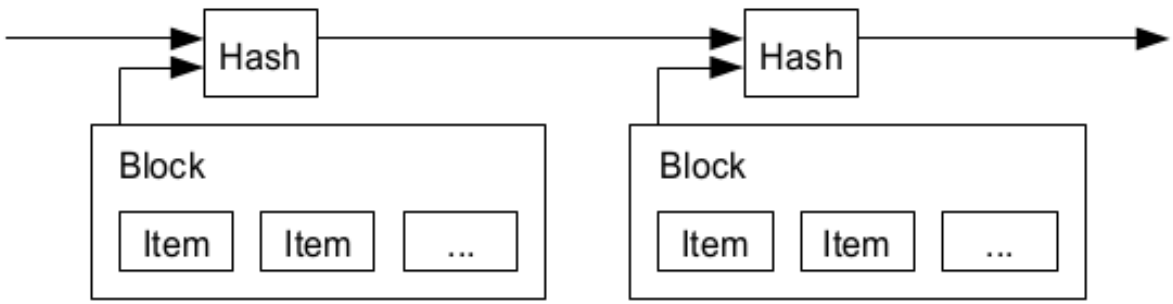
Översikt

Överföringar



Figur : En överföring från Owner 1 till Owner 2 till Owner 3. Bild: [Nak08].

Överföringar

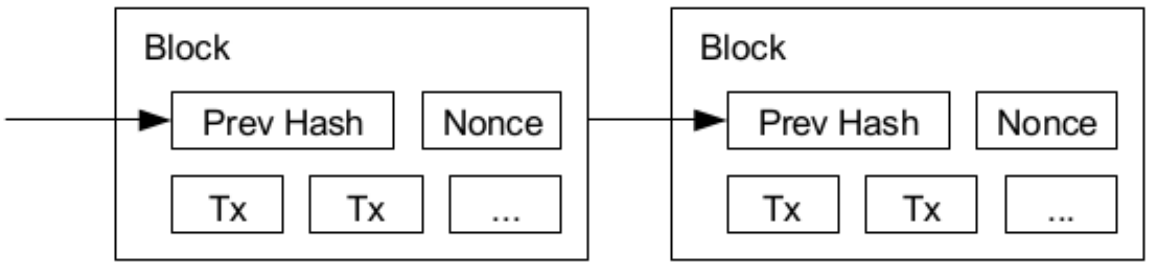


Figur : En enkel tidsstämplingsmekanism. Bild: [Nak08].

Överföringar

- Behöver kunna förhindra förfalskning.
- Den enkla tidsstämplingsmekanismen ger $h_i = h(h_{i-1} \,||\, b_i)$, där h är en kollisionsresistent hashfunktion.
- Denna är enkel att förändra.

Överföringar



Figur : En tidstämplingsmekanism för att förhindra förfalskning. Bild: [Nak08].

Överföringar

- Det vill säga: $h_i = h(h_{i-1} \,||\, b_i \,||\, n)$, där n är en nonce.
- Svårighetsgraden kontrolleras genom att justera antalet inledande nollor i hashvärdet, det vill säga $h_i < k$ är ett krav.
- Detta är att lösa problemet inversa avbildningen.
- Svårighetsgraden k justeras så att ett sådant problem kan lösas var 10:e minut.

Översikt

Decentralisering

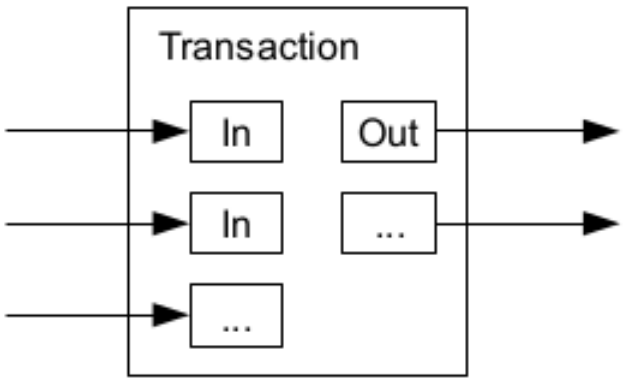
- Varje överföring sänds till alla deltagande noder.
- Varje nod samlar överföringar till ett block.
- Varje nod försöker att finna en nonce n för detta block som uppfyller att $h_i < k$ för nu gällande k.
- När en nod finner detta publiceras det till alla noder i nätverket.
- Detta accepteras av noderna om alla transaktioner i blocket inte finns med tidigare i kedjan och hashvärdena stämmer.
- Att ett block godtas uttrycks genom att noder utgår från detta block när de skapar nästa block i kedjan.

Decentralisering

- Säkerheten kräver att ärliga noder har majoritet.
- Alla noder accepterar den längsta existerande kedjan vid eventuella komplikationer.

Översikt

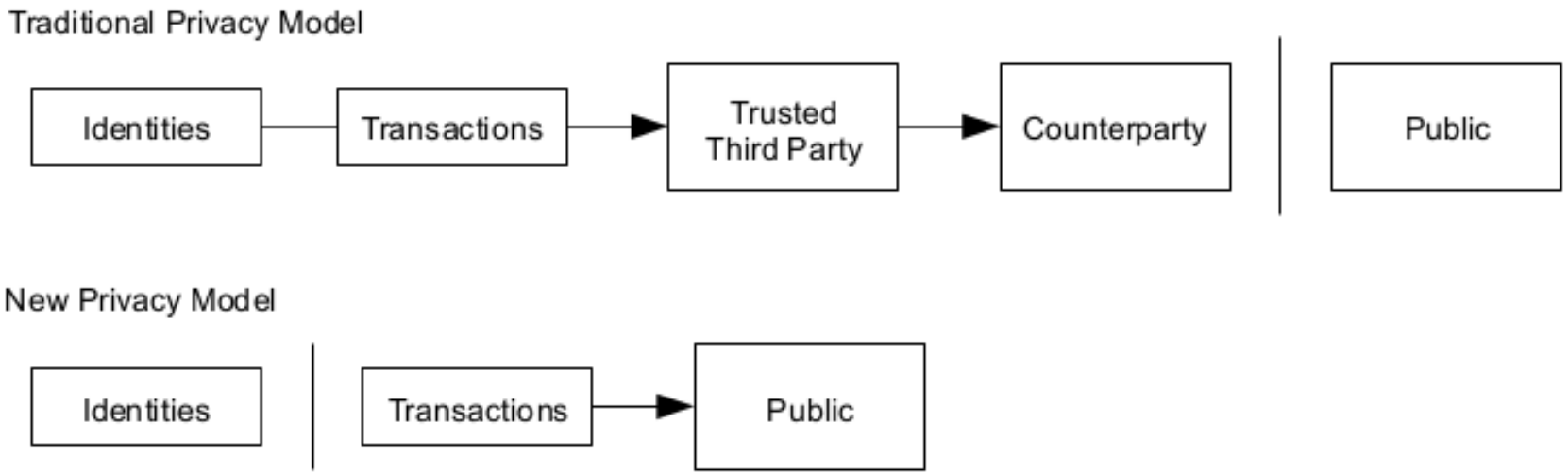
Att dela upp mynt



Figur : En bitcoinöverföring. Bild: [Nak08].

Översikt

Personlig integritet



Figur : Bitcoins modell för anonymitet. Bild: [Nak08].

Referenser I

[And08] Ross J. Anderson. *Security engineering : a guide to building dependable distributed systems*. Wiley, Indianapolis, IN, 2 utgåvan, 2008.

[BBSU12] Simon Barber, Xavier Boyen, Elaine Shi och Ersin Uzun. Bitter to better–how to make bitcoin a better currency. I: *Financial Cryptography and Data Security*, ss 399–414. Springer, 2012.

[Kam13] Dan Kaminsky. Let's cut through the Bitcoin hype: A hacker–entrepreneur's take. Wired, may 2013.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[Pec12] Morgen E. Peck. Bitcoin: The cryptoanarchists' answer to cash. IEEE Spectrum, jun 2012.

[Sta11] William Stallings. *Cryptography and network security : principles and practice*. Prentice Hall, Upper Saddle River, 5. ed., international ed. utgåvan, 2011.