

Kryptografiska mekanismer och valutor

Daniel Bosk¹

Avdelningen för informations- och kommunikationssystem,
Mittuniversitetet, SE-851 70 Sundsvall.

hashsign.tex 2068 2014-11-03 10:52:07Z danbos

¹Detta verk är tillgängliggjort under licensen Creative Commons Erkännande-DelaLika 2.5 Sverige (CC BY-SA 2.5 SE). För att se en sammanfattning och kopia av licenstexten besök URL

<http://creativecommons.org/licenses/by-sa/2.5/se/>

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; Bar+12].

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

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

Asymmetrisk kryptering

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

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.

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 $h_k: \mathcal{X} \rightarrow \mathcal{Y}$.

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 $h: \mathcal{X} \rightarrow \mathcal{Y}$ 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 $h: \mathcal{X} \rightarrow \mathcal{Y}$ 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 $h: \mathcal{X} \rightarrow \mathcal{Y}$.
- 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 $h(x)$ bestäms 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

Algorithm (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 $\mathcal{X}_0 = \{x_1, \dots, x_Q\}$ 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

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

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

Algorithm (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:

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

Formell behandling av hashfunktioner

Bevis.

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

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

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

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

för $2 \leq i \leq Q$. Då har vi att

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

Följaktligen blir sannolikheten för minst en kollision

$$1 - \Pr(E_1 \wedge \dots \wedge E_Q).$$

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, snabbt 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

- 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

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

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.

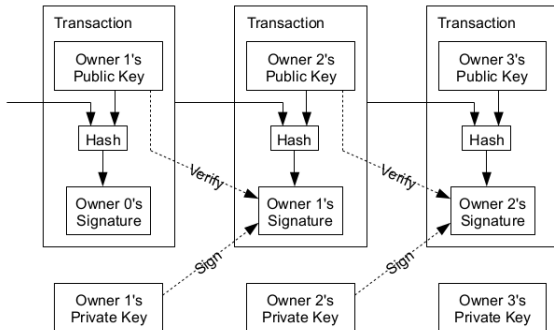
Översikt

- 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

Överföringar



Figur : En överföring från Owner 1 till Owner 2 till Owner 3. 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

- 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.

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.

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

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

Referenser I

- [And08] Ross J. Anderson. *Security Engineering. A guide to building dependable distributed systems*. 2. utg. Indianapolis, IN: Wiley, 2008. ISBN: 978-0-470-06852-6 (hbk.) URL:
<http://www.cl.cam.ac.uk/~rja14/book.html>.
- [Bar+12] 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*. Springer, 2012, s. 399–414.
- [Kam13] Dan Kaminsky. *Let’s Cut Through the Bitcoin Hype: A Hacker–Entrepreneur’s Take*. Wired. Maj 2013. URL:
<http://www.wired.com/opinion/2013/05/lets-cut-through-the-bitcoin-hype/>.

Referenser II

- [Nak08] Satoshi Nakamoto. “Bitcoin: A Peer-to-peer Electronic Cash System”. 2008. URL:
<http://bitcoin.org/bitcoin.pdf>.
- [Pec12] Morgen E. Peck. *Bitcoin: The Cryptoanarchists' Answer to Cash*. IEEE Spectrum. Juni 2012. URL:
<http://spectrum.ieee.org/computing/software/bitcoin-the-cryptoanarchists-answer-to-cash/0>.
- [Sta11] William Stallings. *Cryptography and network security : principles and practice*. 5. ed., International ed. Upper Saddle River: Prentice Hall, 2011. ISBN: 0-13-705632-X (pbk).