Криптографија
- дванаести део -

доц. др Драган Ђокић

Математички факултет, Универзитет у Београду

dragan.djokic@matf.bg.ac.rs

24. мај 2024.

Временски печат

- омогућава да знамо тачно време када је документ настао, тј. да ли је постојао у неком тренутку нпр.
  - Патент због заштите ауторских права

Временски печат

- омогућава да знамо тачно време када је документ настао, тј. да ли је постојао у неком тренутку нпр.
  - Патент због заштите ауторских права
  - Јавно одрицање потписа - Алиса може (анонимно) да објави свој кључ и да затим тврди да је неко други потписао нешто у њено име. Зато је битно упоредити време потписа и време губитка тајности кључа

Временски печат

- омогућава да знамо тачно време када је документ настао, тј. да ли је постојао у неком тренутку нпр.
  - Патент због заштите ауторских права
  - Јавно одрицање потписа - Алиса може (анонимно) да објави свој кључ и да затим тврди да је неко други потписао нешто у њено име. Зато је битно упоредити време потписа и време губитка тајности кључа
- Тома = особа која прави временске печате

Први протокол:

- Алиса шаље Томи хеш документа H
  - хешира се због уштеде меморије и спречавање Цице да види документ

Први протокол:

- Алиса шаље Томи хеш документа H
  - хешира се због уштеде меморије и спречавање Цице да види документ
- Тома враћа Алиси $\left(M^{d_T} \bmod n_T, n_T, e_T\right)$, где је M текст који садржи податке: Алиса, хеш документа H, датум и време t, Томина мрежна адреса..., n_T и e_T су Томини јавни кључеви за RSA

Први протокол:

- Алиса шаље Томи хеш документа H
  - хешира се због уштеде меморије и спречавање Цице да види документ
- Тома враћа Алиси $\left(M^{d_T} \bmod n_T, n_T, e_T\right)$, где је M текст који садржи податке: Алиса, хеш документа H, датум и време t, Томина мрежна адреса..., n_T и e_T су Томини јавни кључеви за RSA
- Тома не чува ништа, неизводљиво је да он има све податке

Први протокол:

- Алиса шаље Томи хеш документа H
  - хешира се због уштеде меморије и спречавање Цице да види документ
- Тома враћа Алиси $\left(M^{d_T} \bmod n_T, n_T, e_T\right)$, где је M текст који садржи податке: Алиса, хеш документа H, датум и време t, Томина мрежна адреса..., n_T и e_T су Томини јавни кључеви за RSA
- Тома не чува ништа, неизводљиво је да он има све податке
- Проблем: Алиса би могла да подмити Тому да јој изда временски печат са другим датумом

Први протокол:

- Алиса шаље Томи хеш документа H
  - хешира се због уштеде меморије и спречавање Цице да види документ
- Тома враћа Алиси $\left(M^{d_T} \bmod n_T, n_T, e_T\right)$, где је M текст који садржи податке: Алиса, хеш документа H, датум и време t, Томина мрежна адреса..., n_T и e_T су Томини јавни кључеви за RSA
- Тома не чува ништа, неизводљиво је да он има све податке
- Проблем: Алиса би могла да подмити Тому да јој изда временски печат са другим датумом
- Ову услугу ради фирма Digistamp, нема пријављених случајева злоупотребе

Други протокол:

- Исто као претходно, с тим да Тома чува сада шаље Алиси $\left(M^{d_T} \bmod n_T, n_T, e_T, k\right)$

Други протокол:

- Исто као претходно, с тим да Тома чува сада шаље Алиси $\left(M^{d_T} \bmod n_T, n_T, e_T, k\right)$
- k је редни број временског печата. Периодично (нпр. једном дневно) Тома објављује последњи искоришћени k

Други протокол:

- Исто као претходно, с тим да Тома чува сада шаље Алиси $\left(M^{d_T} \bmod n_T, n_T, e_T, k\right)$
- k је редни број временског печата. Периодично (нпр. једном дневно) Тома објављује последњи искоришћени k
- Тома објављује све потписе и они се уписују у више евиденција
  - захтева велики складишни простор
  - Тома нема овлашћења да мења те евиденције како не би пао у искушење да мења старе временске печате

Блокчејн технологија

Алиса уплаћује новац Бобану

- Банка евидентира да је Алиса потрошила новац и да Бобан добио новац, не зна се да је трансакција између њих

Блокчејн технологија

Алиса уплаћује новац Бобану

- Банка евидентира да је Алиса потрошила новац и да Бобан добио новац, не зна се да је трансакција између њих
- Банка има комплетну евиденцију свих промена на рачуну корисника

Блокчејн технологија

Алиса уплаћује новац Бобану

- Банка евидентира да је Алиса потрошила новац и да Бобан добио новац, не зна се да је трансакција између њих
- Банка има комплетну евиденцију свих промена на рачуну корисника
- Блокчејн нуди већи приватност
  - Идентитет је сакривен, јавни су само износи

Блокчејн технологија

Алиса уплаћује новац Бобану

- Банка евидентира да је Алиса потрошила новац и да Бобан добио новац, не зна се да је трансакција између њих
- Банка има комплетну евиденцију свих промена на рачуну корисника
- Блокчејн нуди већи приватност
  - Идентитет је сакривен, јавни су само износи
  - Не постоји банка, нема централне контроле и евиденције.

Блокчејн технологија

Алиса уплаћује новац Бобану

- Банка евидентира да је Алиса потрошила новац и да Бобан добио новац, не зна се да је трансакција између њих
- Банка има комплетну евиденцију свих промена на рачуну корисника
- Блокчејн нуди већи приватност
  - Идентитет је сакривен, јавни су само износи
  - Не постоји банка, нема централне контроле и евиденције.
  - Одређени учесници у систему (чворови) одлучују која трансакција је исправна. Има више хиљада чворова

Блокчејн технологија

Алиса уплаћује новац Бобану

- Банка евидентира да је Алиса потрошила новац и да Бобан добио новац, не зна се да је трансакција између њих
- Банка има комплетну евиденцију свих промена на рачуну корисника
- Блокчејн нуди већи приватност
  - Идентитет је сакривен, јавни су само износи
  - Не постоји банка, нема централне контроле и евиденције.
  - Одређени учесници у систему (чворови) одлучују која трансакција је исправна. Има више хиљада чворова
  - Чворови праве своје копије (дела) евиденције.

- Основна примена блокчејна јесте дигитални новац нпр. биткоин, али се користи и за
  - дигиталне уговоре
  - електронско гласање
  - заштиту ауторских права...

- Основна примена блокчејна јесте дигитални новац нпр. биткоин, али се користи и за
  - дигиталне уговоре
  - електронско гласање
  - заштиту ауторских права...
- Протокол је осмислио Сатоши Накамото (псеудоним), почетна вредност биткоина 2009. је била мања од једног динара, данас вреди око $7,5$ милиона динара

- Основна примена блокчејна јесте дигитални новац нпр. биткоин, али се користи и за
  - дигиталне уговоре
  - електронско гласање
  - заштиту ауторских права...
- Протокол је осмислио Сатоши Накамото (псеудоним), почетна вредност биткоина 2009. је била мања од једног динара, данас вреди око $7,5$ милиона динара
- Биткоин користи
  - ЕлГамалов дигитални потпис са елиптичком кривом $E:\ y^2 = x^3 + 7$ над пољем $\mathbb{Z}_p$, где је p прост 256-битни број и G генератор $E\left(\mathbb{Z}_p\right)$ (p и G су фиксирани и познати)
  - поновљени SHA256 алгоритам за хеширање

- Основна примена блокчејна јесте дигитални новац нпр. биткоин, али се користи и за
  - дигиталне уговоре
  - електронско гласање
  - заштиту ауторских права...
- Протокол је осмислио Сатоши Накамото (псеудоним), почетна вредност биткоина 2009. је била мања од једног динара, данас вреди око $7,5$ милиона динара
- Биткоин користи
  - ЕлГамалов дигитални потпис са елиптичком кривом $E:\ y^2 = x^3 + 7$ над пољем $\mathbb{Z}_p$, где је p прост 256-битни број и G генератор $E(\mathbb{Z}_p)$ (p и G су фиксирани и познати)
  - поновљени SHA256 алгоритам за хеширање
- Алиса уплаћује Бобану новац тако што саставе документ (трансакцију) који потписује Алиса, а Бобан је шаље свим чворовима.
  - Трансакција мора да садржи хеш претходне трансакције (или више њих) у којој је Алиса добила новац који сад уплаћује Бобану

Пример трансакције:

Пример 20.2. *Нека је $tran_q$ нека од претходних трансакција. Алиса (A) је примила том трансакцијом износ 100. (Трансакције уствари нису нумерисане; у примеру је згодно имати њихове редне бројеве.) A жели да уплати 100 Бобану (B). B шаље A хеш $hash(P_{B,r})$. Приметимо да су $P_{A,q} = n_{A,q}G$ и $P_{B,r} = n_{B,r}G$ тачке — јавни кључеви. Корисник користи нову тачку — јавни кључ за сваку трансакцију.*

Ово је нешто упрошћена трансакција $tran_r$:

a. $hash(ST_{q,0})$ // *$ST_{q,0}$ ис а упрошћена верзија* $tran_q$

b. 0 // *индекс у трансакцији* $tran_q$ *износа* 100

c. $P_{A,q}$

d. $potpis(n_{A,q}, hash(ST_{r,0}))$ // $ST_{r,0} = abcef$ *из* $tran_r$

e. 100 // *вредност са индексом* 0 *у* $tran_r$

f. $hash(P_{B,r})$

g. $hash(ST_{r,0})$ // *ова хеш вредност завршава* $tran_r$.

Приметимо да $tran_r$ садржи информацију о преносу износа од A ка B.
B шаље $tran_r$ свим чворовима у мрежи.

Сваки чвор у мрежи

- *проверава $hash(ST_{q,0})$ у делу a. трансакције $tran_r$ у односу на запис у бази података; циљ је да већина чворова то прихвати. Другим речима, проверава се да се део a. трансакције $tran_r$ слаже са делом g. претходно прихваћене трансакције $tran_q$.*
- *проверава да ли је износ на индексу 0 у $tran_q$ већи или једнак од 100 (део e трансакције $tran_r$).*
- *хешира $P_{A,q}$ из дела c трансакције $tran_r$ и упоређује резултат са $hash(P_{A,q})$ у трансакцији $tran_q$.*
- *хешира $ST_{r,0} = abcef$ у трансакцији $tran_r$ и упоређује резултат са g у трансакцији $tran_r$.*
- *израчунава $potpis^{-1}(P_{A,q}, d)$ (проверава потпис) и упоређује то са g (обе ствари су у $tran_r$).*

- Нигде у трансакцији се не појављују имена

- Нигде у трансакцији се не појављују имена
- Корисник чува своје тајне кључеве из свих трансакција у којима је примио биткоине, све док их не искористи за слање биткоина

- Нигде у трансакцији се не појављују имена
- Корисник чува своје тајне кључеве из свих трансакција у којима је примио биткоине, све док их не искористи за слање биткоина
- Бобан шаље трансакцију свим чворовима и они је уписују у своју евиденцију након што је провере

- Нигде у трансакцији се не појављују имена
- Корисник чува своје тајне кључеве из свих трансакција у којима је примио биткоине, све док их не искористи за слање биткоина
- Бобан шаље трансакцију свим чворовима и они је уписују у своју евиденцију након што је провере
- Ситуација из претх. примера (Алиса уплаћује тачно онолико колико је раније добила) је ретка

- Нигде у трансакцији се не појављују имена
- Корисник чува своје тајне кључеве из свих трансакција у којима је примио биткоине, све док их не искористи за слање биткоина
- Бобан шаље трансакцију свим чворовима и они је уписују у своју евиденцију након што је провере
- Ситуација из претх. примера (Алиса уплаћује тачно онолико колико је раније добила) је ретка
- У пракси
  - Алиса може да споји новац из више трансакција и плати са тим

- Нигде у трансакцији се не појављују имена
- Корисник чува своје тајне кључеве из свих трансакција у којима је примио биткоине, све док их не искористи за слање биткоина
- Бобан шаље трансакцију свим чворовима и они је уписују у своју евиденцију након што је провере
- Ситуација из претх. примера (Алиса уплаћује тачно онолико колико је раније добила) је ретка
- У пракси
  - Алиса може да споји новац из више трансакција и плати са тим
  - Може да има више прималаца, ту су укључени и
    - чвор који ће узети провизију за трансакцију
    - Алиса, која ће уплатити сама себи вишак (више не може да користи искоришћене трансакције, ако је нешто преостало позиваће се на ову нову трансакцију)

У оквиру трансакције $tran_s$ *B жели да искористи износ* 100 *из* $tran_r$ *и износ* 20 *из претходне трансакције* $tran_p$*, да уплати износ* 110 *учеснику C, и да износ* 10 *уплати назад самом себи.*

C шаље B поруку $hash(P_{C,s})$ *("адресу" свог биткоин новчаника, односно своју јавну тачку).*

Ово је трансакција $tran_s$*:*

a. $hash(ST_{r,0})$ *// g из трансакције* $tran_r$

b. 0 *// индекс из e у трансакцији* $tran_r$*, који показује на износ* 100

c. $P_{B,r}$

d. $potpis(n_{B,r}, hash(ST_{s,0}))$ *//* $ST_{s,0} = abcij$ *из трансакције* $tran_s$

e. $hash(ST_{p,0})$

f. 0 *// индекс из трансакције* $tran_p$*, који показује на износ* 20

g. $P_{B,p}$

h. $potpis(n_{B,p}, hash(ST_{s,1}))$ *//* $ST_{s,1} = efglm$ *из трансакције* $tran_s$

i. 110 *// индекс* 0 *за* $tran_s$

j. $hash(P_{C,s})$

k. $hash(ST_{s,0})$

l. 10 *// индекс* 1 *за* $tran_s$

m. $hash(P_{B,s})$ *// B креира јавну тачку — биткоин адресу* $P_{B,s}$ *на коју очекује износ* 10*.*

n. $hash(ST_{s,1})$ *// Ово закључује трансакцију* $tran_s$*.*

Учесници B и C (пошто су обоје примаоци уплата) шаљу $tran_s$ *свим чворовима у мрежи. Као и за претходну трансакцију, чворови проверавају хеш вредности, износе, потписе, слично као за трансакцију* $tran_r$*.*

- Блок садржи
  - листу појединачних трансакција (највише 2400)
  - хеш вредност претходног блока
  - временски печат
  - nonce - случајно генерисан 32-битни број (нема превод)
  - заштитни број T
  - хеш вредност h претходних ставки (256-битна)

- Блок садржи
  - листу појединачних трансакција (највише 2400)
  - хеш вредност претходног блока
  - временски печат
  - nonce - случајно генерисан 32-битни број (нема превод)
  - заштитни број T
  - хеш вредност h претходних ставки (256-битна)
- Дакле, блок је везан за претходни блок и тако се добија ланац блокова - блокчејн

- Блок садржи
  - листу појединачних трансакција (највише 2400)
  - хеш вредност претходног блока
  - временски печат
  - nonce - случајно генерисан 32-битни број (нема превод)
  - заштитни број T
  - хеш вредност h претходних ставки (256-битна)
- Дакле, блок је везан за претходни блок и тако се добија ланац блокова - блокчејн
- Чворови праве блокове и сами одлучују које трансакције укључују у блок нпр. на основу провизије

- Блок је успешно направљен (заштићен) када је $h < T$, тј. h почиње одређеним бројем нула

- Блок је успешно направљен (заштићен) када је $h < T$, тј. h почиње одређеним бројем нула
  - Чвор мења nonce и тиме варира h све док не погоди хеш вредност $h < T$ - ради се грубом силом

- Блок је успешно направљен (заштићен) када је $h < T$, тј. h почиње одређеним бројем нула
  - Чвор мења nonce и тиме варира h све док не погоди хеш вредност $h < T$ - ради се грубом силом
  - Сви чворови покушавају да направе блок, онај ко први успе добија провизије од свих трансакција и награду за успешно направљен блок (рударење)
    - За неке чворове је то нерешив проблем јер је хеш вредност дужа од nonce-a

- Блок је успешно направљен (заштићен) када је $h < T$, тј. h почиње одређеним бројем нула
  - Чвор мења nonce и тиме варира h све док не погоди хеш вредност $h < T$ - ради се грубом силом
  - Сви чворови покушавају да направе блок, онај ко први успе добија провизије од свих трансакција и награду за успешно направљен блок (рударење)
    - За неке чворове је то нерешив проблем јер је хеш вредност дужа од nonce-а
  - Број T се повремено мења, подешава се тако да за креирање блока треба око 10 минута

- Блок је успешно направљен (заштићен) када је $h < T$, тј. h почиње одређеним бројем нула
  - Чвор мења nonce и тиме варира h све док не погоди хеш вредност $h < T$ - ради се грубом силом
  - Сви чворови покушавају да направе блок, онај ко први успе добија провизије од свих трансакција и награду за успешно направљен блок (рударење)
    - За неке чворове је то нерешив проблем јер је хеш вредност дужа од nonce-а
  - Број T се повремено мења, подешава се тако да за креирање блока треба око 10 минута
- Једини начин за додавање нових биткоина у систем је рударење
  - Да би се спречила инфлација: Након сваких 210000 блокова награда се преполовљује, почело се са $50\,BTC$, сада је $3,125\,BTC$. Укупан број биткоина неће прећи 21 милион (личи на геометријски ред)

- Када један чвор објави да је креирао успешан блок, остали чворови провере блок и уписују га у своју евиденцију

- Када један чвор објави да је креирао успешан блок, остали чворови провере блок и уписују га у своју евиденцију
- Може да се деси
  - да више чворова (приближно) истовремено објави блок - онда неки чворови прихвате један блок, неки прихвате други
  - злонамерни чвор подметне блок који садржи неисправне трансакције (нпр. двоструко плаћање истим биткоинима) и који прихвате неки злонамерни чворови

- Када један чвор објави да је креирао успешан блок, остали чворови провере блок и уписују га у своју евиденцију
- Може да се деси
  - да више чворова (приближно) истовремено објави блок - онда неки чворови прихвате један блок, неки прихвате други
  - злонамерни чвор подметне блок који садржи неисправне трансакције (нпр. двоструко плаћање истим биткоинима) и који прихвате неки злонамерни чворови
- Ланац је настављен на више начина, прихвата се онај ланац који има више надовезаних блокова - 6 је довољна гаранција
  - Зато се биткоини добијени трансакцијом не могу одмах користити

- Када један чвор објави да је креирао успешан блок, остали чворови провере блок и уписују га у своју евиденцију
- Може да се деси
  - да више чворова (приближно) истовремено објави блок - онда неки чворови прихвате један блок, неки прихвате други
  - злонамерни чвор подметне блок који садржи неисправне трансакције (нпр. двоструко плаћање истим биткоинима) и који прихвате неки злонамерни чворови
- Ланац је настављен на више начина, прихвата се онај ланац који има више надовезаних блокова - 6 је довољна гаранција
  - Зато се биткоини добијени трансакцијом не могу одмах користити
- Заштита од преваре: Да би неки чворови подметнули неисправну трансакцију морају да буду бржи од свих осталих заједно
  - Већина чворова ради исправно како би зарадили рударењем

Етереум

За разлику од биткоина

- има слабије захтеве за заштиту блока, блокови се формирају за око 15 секунди
- не смањује награде за рударење, опасност од инфлације

Етереум

За разлику од биткоина

- има слабије захтеве за заштиту блока, блокови се формирају за око 15 секунди
- не смањује награде за рударење, опасност од инфлације

Етереум паметни уговори

- **Етереум** омогућава да се у програмском језику **Solidity** програмирају **“паметни уговори”**, који се аутоматски извршавају на блокчејну
  - На пример, инвеститори желе да помогну развој неког пројекта (crowdfunding)
  - Пројекат ће се покренути ако се скупи довољно инвестиција
  - Паметни уговор се поставља тако да сви заинтересовани инвеститори преносе свој “новац” у блокчејн и тај новац је намењен искључиво за тај пројекат
  - Ако се до одређеног датума сакупи довољно инвестиција, сав “новац” ће бити на располагању организаторима пројекта
  - Ако се не скупи довољно инвестиција, сав “новац” ће бити враћен инвеститорима
    - организатор пројекта има гаранцију да инвеститори не могу да “се предомисле”
    - инвеститори пројекта имају гаранцију да ће им новац бити враћен ако се не скупи довољно инвестиција

Дељење тајне

- Неку тајну S треба поделити између n особа, у које немамо потпуно поверење

Дељење тајне

- Неку тајну S треба поделити између n особа, у које немамо потпуно поверење
- Лоша решења су да
  - свако зна целу тајну - неко може да ода тајну
  - поделити тајну на делове $S_1, S_2, \ldots, S_n$, тако да се цела тајна може реконструисати само ако су познати сви S_i-ови. Тада неко може да одбије да нам достави свој део тајне S_i, чиме је тајна S изгубљена

Дељење тајне

- Неку тајну S треба поделити између n особа, у које немамо потпуно поверење
- Лоша решења су да
  - свако зна целу тајну - неко може да ода тајну
  - поделити тајну на делове $S_1, S_2, \ldots, S_n$, тако да се цела тајна може реконструисати само ако су познати сви S_i-ови. Тада неко може да одбије да нам достави свој део тајне S_i, чиме је тајна S изгубљена
- Најбоље би било да је за реконструкцију тајне S довољно познавање делова које поседује тачно m особа, где је $m \in [1, n]$ оптимално изабрано

Шамиров поступак:

- Нека је S тајна кодирана са l битова

Шамиров поступак:

- Нека је S тајна кодирана са l битова
- Бира се случајан прост број $p > 2^l$ и позитивни цели бројеви $a_1, a_2, \ldots, a_{m-1} < p$. Нека је $f(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots a_1 x + S$

Шамиров поступак:

- Нека је S тајна кодирана са l битова
- Бира се случајан прост број $p > 2^l$ и позитивни цели бројеви $a_1, a_2, \ldots, a_{m-1} < p$. Нека је $f(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots a_1 x + S$
- i-тој особи треба дати $S_i = f(i) \bmod p$

Шамиров поступак:

- Нека је S тајна кодирана са l битова
- Бира се случајан прост број $p > 2^l$ и позитивни цели бројеви $a_1, a_2, \ldots, a_{m-1} < p$. Нека је $f(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots a_1 x + S$
- i-тој особи треба дати $S_i = f(i) \bmod p$

Рекострукција тајне:

- Када је познато m тајни S_i, формирамо систем (над пољем $\mathbb{Z}_p$) од m једначина по променљивим $S, a_1, a_2, \ldots, a_{m-1}$

Шамиров поступак:

- Нека је S тајна кодирана са l битова
- Бира се случајан прост број $p > 2^l$ и позитивни цели бројеви $a_1, a_2, \ldots, a_{m-1} < p$. Нека је $f(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots a_1 x + S$
- i-тој особи треба дати $S_i = f(i) \bmod p$

Рекострукција тајне:

- Када је познато m тајни S_i, формирамо систем (над пољем $\mathbb{Z}_p$) од m једначина по променљивим $S, a_1, a_2, \ldots, a_{m-1}$
- Детерминанта тог система је Вандермондова детерминанта која не може бити нула, па систем има јединствено решење

Пример: Поделити тајну 401 између пет особа, тако да сваке три особе могу да реконструишу тајну

Тајни број је 401. *Бира се прост број* $p = 587$, $a = 322$, $b = 149$. *Према томе* $f(x) = 322x^2 + 149x + 401 \pmod{587}$. *Тада је* $f(1) = 285$, $f(2) = 226$, $f(3) = 224$, $f(4) = 279$, $f(5) = 391$. *Први, четврти и пети менаџери могу да се договоре да заједно добију тајну* S. *Они знају* $f(x) = ax^2 + bx + S \pmod{587}$, *али не знају* a, b, S. *Они знају* $f(1) = a + b + S = 285$, $16a + 4b + S = 279$ *и* $25a + 5b + S = 391$. *Они решавају систем*

$$\begin{bmatrix} 1 & 1 & 1 \\ 16 & 4 & 1 \\ 25 & 5 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ S \end{bmatrix} = \begin{bmatrix} 285 \\ 279 \\ 391 \end{bmatrix} \pmod{587}$$

да би одредили a, b, S. *Њих занима само* S.