Sokea RSA-allekirjoitus

- Toinen tapa allekirjoittaa sanomaa näkemättä.
- Käytetään RSA-allekirjoitusta, jossa sanoma kerrotaan satunnaisella kertoimella.

  Trentin julkinen avain on (e, n) ja salainen (d, n). Alicen sanoma on m.

  1. Alice generoi satunnaisluvun k ja sekoittaa sen avulla sanoman,
  $$t = m\,k^e \bmod n.$$
  2. Trent allekirjoittaa sekoitetun sanoman,
  $$t^d \bmod n = (m\,k^e)^d \bmod n = m^d k \bmod n.$$
  3. Alice poistaa sekoituskertoimen,
  $$s = t^d k^{-1} \bmod n = m^d \bmod n.$$
- **Etu**: Trent ei koskaan näe asiakirjaa, eikä myöhemmin pysty yhdistämään sitä tähän allekirjoitustapahtumaan ja Aliceen.

- Käytännössä lasketaan sanomasta ensin sormenjälki.

  Trentin julkinen avain on (e, n) ja salainen (d, n). Alicen sanoma on m.

  1. Alice laskee sanomasta sormenjäljen,
  $$H = h(m).$$
  2. Alice generoi satunnaisluvun k ja sekoittaa sen avulla sanoman,
  $$t = H\,k^e \bmod n.$$
  3. Trent allekirjoittaa sekoitetun sanoman,
  $$t^d = (H\,k^e)^d \bmod n = H^d k \bmod n.$$
  4. Alice poistaa sekoituskertoimen,
  $$s = t^d k^{-1} \bmod n = H^d \bmod n.$$
  5. Alice liittää allekirjoituksen sanomaan
  $$m, s.$$
- Käytetään anonyymissä sähköisessä rahassa. Allekirjoittaja todistaa aitouden, muttei tunnista seteliä myöhemmin.

Cut and choose

- **Ongelma**: Trent haluaa valvoa allekirjoittamansa sanoman sisältöä, vaikkei saakaan tietää sitä täsmälleen.
- Esim. Alice arpoo satunnaisluvun ja lisää siihen aikaleiman. Luotettu Trent allekirjoittaa luvun, jottei Alice voi muuttaa sitä myöhemmin. Trent ei saa nähdä lukua, mutta hänen pitää taata, että aikaleima on oikea.
- Propabilistinen menetelmä (*cut and choose*) :

  1. Alice arpoo sata lukua X, kirjoittaa ne paperilapuille: T_A, X.
  2. Alice sulkee laput kirjekuoriin ja lähettää kirjekuoret Trentille.
  3. Trent repii auki 99 satunnaisesti valitsemaansa kirjekuorta ja lukee sisällön.
  4. Jos 99 sanomaa olivat kunnossa, Trent päättelee viimeisenkin olevan. Hän allekirjoittaa kirjekuoren ja lähettää Alicelle.

- Sama sähköisesti:

  1. Alice arpoo 100 lukua X, kirjoittaa ne sanomiksi (T_A, X), laskee sormenjäljet ja sekoittaa satunnaisilla kertoimilla.
  2. Alice lähettää sekoitetut sanomat Trentille allekirjoitettavaksi.
  3. Trent pyytää Alicelta 99 alkuperäistä sanomaa nähtäväkseen sekoituskertoimen kera.
  4. Jos 99 sanomaa on kunnossa, Trent päättelee viimeisenkin olevan. Hän allekirjoittaa sen ja lähettää sen Alicelle.
- Alice onnistuu virheellisen sanoman kirjoittamisessa todennäköisyydellä $1/100$.

  Kiinni jäämisestä oltava riittävät sanktiot. Esim. huijari ei enää saa osallistua.
- Jos huijauksen onnistumistodennäköisyys on hyvin pieni, ei kiinni jäämisestä tarvitse rangaista.

  Ratkaisu: n:stä sanomasta tarkistetaan puolet, huijauksen onnistumiseksi loppujen pitää olla vääriä. Huijauksen onnistumistodennäköisyys
  $$\frac{(n/2)!^2}{n!}.$$

Anonyymi elektroninen raha

- Alice on maksaja, Bob maksun saaja.
  Pankin julkinen RSA-avain on $(3, n)$ ja salainen (d, n).

  1. Alice arpoo sarjanumeron ja satunnaisluvun R. SN = setelin sarjanumero yhdistettynä allekirjoituksen vaatimaan redundanttiin dataan. Alice lähettää pankille
     $$SN \cdot R^3 \bmod n.$$
  2. Pankki allekirjoittaa sanoman salaisella avaimellaan ja lähettää Alicelle
     $$(SN \cdot R^3)^d \bmod n \;=\; SN^d \cdot R \bmod n.$$
  3. Alice jakaa R:n pois ja muodostaa setelin
     $$(SN,\; SN^d \bmod n).$$

- Markkamäärä ei lue setelissä, vaan erikokoiset rahat allekirjoitetaan eri avaimella ($e = 3, 5, 7, \ldots$).
- Pankki ei pysty yhdistämään seteliä Aliceen. Entä Bob?
- Setelin kaksinkertainen käyttö huomataan sarjanumerosta.
  Ongelma: Kuinka Alice saadaan kiinni?

Ecash

- David Chaum 1988
- Anonyymi raha, jossa kaksinkertainen käyttäjä paljastuu.
- Idea: jaetaan nimi "Alice" kahteen osaan, jotka yhdessä paljastavat nimen.
  $$a, \quad a \oplus \text{"Alice"}.$$
  Bob valitsee satunnaisesti, kumman osan Alice kertoo.
- Pankki allekirjoittaa setelin sokeasti, mutta haluaa varmistua, että setelissä olevat osat yhdessä todella tuottavat nimen "Alice". Käytetään *cut and choose*-menetelmää.

- Ecash-setelin luonti
  1. Alice generoi 200 satunnaislukua a_i ja laskee sormenjäljet
     $$x_i = h(a_i), \quad y_i = h(a_i \oplus \text{"Alice"}).$$
  2. Alice generoi 200 satunnaislukua R_i ja lähettää Trentille sanomat
     $$h(x_i, y_i) \cdot R_i^3 \bmod n.$$
  3. Trent pyytää Alicelta 100 paria satunnaislukuja R_i ja a_i ja tarkistaa, ettei Alice yritä huijata. (*cut and choose*)
  4. Trent allekirjoittaa loput 100 sormenjälkeä ja lähettää Alicelle
     $$\big(h(x_i, y_i) \cdot R_i^3\big)^d \bmod n \;=\; h(x_i, y_i)^d \cdot R_i \bmod n.$$
  5. Alice jakaa pois R_i:t,
     $$h(x_i, y_i)^d \bmod n.$$
- Seteli on muotoa
  $$S = \prod_{j=1}^{100} h(x_j, y_j)^d.$$
- Huijauksen onnistumistodennäköisyys on $\frac{(50!)^2}{100!} \approx 2^{-96}$.

- Ecash-maksu
  1. Alice lähettää Bobille setelin S.
  2. Bob generoi 64 satunnaista bittiä, liittää ne 36-bittiseen myyjätunnukseensa ja lähettää Alicelle.
  3. Jos js bitti on 0, Alice lähettää Bobille
     $$x_j,\; a_j \oplus \text{"Alice"}.$$
     Jos bitti on 1, Alice lähettää Bobille
     $$y_j,\; a_j.$$
  4. Bob tarkistaa, että
     $$S^3 = \prod_{j=1}^{100} h(x_j, y_j).$$
- Bob ei saa selville Alicen nimeä, sillä hän tietää parista $(a_j,\; a_j \oplus \text{"Alice"})$ vain toisen osan.
- Myöhemmin Bob lähettää pankkiin setelin S ja Alicelta saamansa luvut.
  Jos Alice käyttää setelin kahteen kertaan ja Bob valitsee eri 64 satunnaista bittiä, paljastuu Alice. Huijauksen onnistumistodennäköisyys on 2^{-64}
- Hyvä idea, mutta todellisessa Ecash-järjestelmässä tuplakäyttö tarkistetaan pankista *on-line*!

Mikromaksut

(*micropayments*)

- Halutaan maksaa pieniä summia (10 penniä - 10 markkaa) esim. WWW-sivuista ja minuuttiveloitteisista TV-kanavista.

  Tavalliset maksutavat ovat liian hitaita. Maksusuoritus on kalliimpi kuin ostettava palvelu.
- **Mikromaksut** on optimoitu laskennan, muistin käytön ja kommunikaation kannalta tehokkaiksi ja halvoiksi.
- Riittävä turvallisuus:
  - huijaaminen kalliimpaa kuin hyöty
  - toistuva väärinkäyttäjä jää kiinni
  - rahan katoaminen virhetilanteessa ei haittaa
  - osittainen anonymiteetti perustuu logien keräämisen kalleuteen.
- Vältetään
  - julkisen avaimen algoritmeja
  - *on-line* -laskentaa ja yhteyttä pankkiin
  - muistin tuhlausta.

  Suositaan
  - yksisuuntaisia hajautusfunktioita
  - *off-line* -toimintoja.

PayWord

- Rivest, Shamir 1996 (vastaavia Anderson et al. 1995, Pedersen 1995)
- Idea Lamportin vaihtuvista salasanoista. Raha on muotoa

$$h^{100}(rand) \;=\; \underbrace{h(h(\ldots(h(}_{100\ \mathrm{kpl}}rand))\ldots).$$

  Paljastetaan vähitellen $h^{99}(rand)$, $h^{98}(rand)$, ... , $h(rand)$, $rand$.

  h on yksisuuntainen hajautusfunktio, esim. SHA.
- Alicen (ostaja) ja Trentin (pankki, luottofirma) valmistelut kerran kuukaudessa:

  1. Trent kirjoittaa Alicelle sertifikaatin

  $$T,\ addr_A,\ K_A^+,\ ExpTime,\ S_T(\ldots)$$

  T = Trentin nimi
  $addr_A$ = Alicen verkko-osoite
  K_A^+ = Alicen julkinen avain
  $ExpTime$ = sertifikaatin vanhenemisaika
  $S_T(\ldots)$ = Trentin allekirjoitus

  Peruutuslista luottokelvottomista asiakkaista jaetaan myyjille päivittäin.

- Alicelta (ostaja) ja Bobin (myyjä) valmistelut ennen päivän ensimmäistä ostosta:

  1. Alice arvioi Bobilta päivän aikana tekemiensä ostosten määrän N.
  2. Alice generoi satunnaisluvun w_N ja laskee

     $$w_i = h(w_{i+1}),\ i = N-1\ldots0.$$

     Nämä ovat salaisia paitsi w_0.
  3. Alice lähettää Bobille Trentin allekirjoittaman sertifikaatin sekä itse allekirjoittamansa maksulupauksen

     $$B,\ T,\ w_0,\ Date,\ S_A(\ldots)$$

     B = Bobin nimi
     T = Trentin nimi
     $w_0 = h^N(w_N)$
     $Date$ = päivämäärä
     $S_A(\ldots)$ = Alicen allekirjoitus
  4. Bob tarkistaa Trentin ja Alicen allekirjoitukset ja muistaa w_0:n.

- Alice maksaa Bobille 10 penniä:

  1. Alice lähettää Bobille parin

     $$w_i,\ i$$

     missä ensimmäisellä kerralla $i = 1$ ja seuraavilla $i = 2, 3, 4, \ldots, N$.
  2. Bob tarkistaa, että

     $$h(w_i) = w_{i-1}.$$

     ja muistaa uusimman w_i.
- 30 penniä maksetaan lähettämällä $w_{i+3}, i+3$.
- Tilien selvitys päivän päätyttyä:

  1. Bob lähettää Trentille Alicen allekirjoittaman maksulupauksen sekä

     $$w_i,\ i$$

     suurimmalle tuntemalleen i.
  2. Trent veloittaa $i \times 10$ penniä Alicelta ja maksaa sen Bobille.
- Bobin täytyy muistaa Alicen maksulupaukset päivän ajan, jottei Alice käytä niitä uudestaan.

Sinetöity päätös

(*bit commitment*)

- Alice päättää salaisen sanoman, mutta paljastaa sen Bobille vasta myöhemmin. Alice ei saa muuttaa sanomaa kerran sen päätettyään.

  Alicen päätös:

  1. Alice generoi (vakiopituisen) satunnaisen luvun R.
  2. Alice päättää salaisen sanoman M.
  3. Alice laskee yksisuuntaisella funktiolla h ja lähettää Bobille
  $$h(R, M).$$

  Päätöksen paljastaminen:

  1. Alice lähettää Bobille
  $$(R, M).$$
  2. Bob laskee $h(R, M)$ ja vertaa tulosta Alicelta aiemmin saamaansa.

  Huom: kuinka estetään Alicea kopioimasta muiden sanomia niiden sisältöä tietämättä?

Lantinheitto

- Alice ja Bob haluavat heittää lanttia (vetää pitkää tikkua) eli päättää satunnaisesta bitistä ilman, että kumpikaan voi vaikuttaa tulokseen.
- Lantinheitto ilman lanttia:
  1. Alice kirjoittaa paperilapulle "kruuna" tai "klaava" ja sulkee lapun kirjekuoreen.
  2. Bob valitsee "sama" tai "eri".
  3. Kuori avataan. Jos Bob sanoi "sama", on tulos se, mikä lapussa lukee. Jos Bob sanoi "eri", on tulos päinvastainen kuin lapussa lukee.
- Hajautettu lantinheitto:
  1. Alice valitsee 64-bittisen satunnaisluvun R ja bitin b ja lähettää Bobille
  $$h(R, b).$$
  2. Bob valitsee bitin b' ja lähettää sen Alicelle.
  3. Alice lähettää Bobille (R, b).
  4. Tulos on $b \oplus b'$.

Pelikorttien jako

- Käytetään kommutatiivista salausta. (E_A = salaus Alicen salaisella avaimella.)
  1. Alice ja Bob generoivat kumpikin oman salaisen avaimen (tai avainparin).
  2. Alice salaa korttien nimet $Kortti_i$ (riittävästi redundanssia) ja lähettää Bobille arvotussa järjestyksessä
  $$E_A(Kortti_i),\ i = 1 \ldots 52.$$
  3. Bob arpoo 5 korttia Alicen kädeksi ja lähettää Alicelle.
  4. Bob arpoo 5 korttia omaksi kädekseen, salaa ne ja lähettää Alicelle
  $$E_B(E_A(Kortti_{i_k})),\ k = 1 \ldots 5.$$
  5. Alice avaa sanomat ja lähettää Bobille
  $$D_A(E_B(E_A(Kortti_{i_k}))) = E_B(Kortti_{i_k}),\ k = 1...5.$$
  6. Bob avaa kätensä salaisella avaimellaan: $Kortti_{i_k},\ k = 1 \ldots 5$.
  7. Pelin päätyttyä Alice ja Bob paljastavat avaimensa. Kumpikin tarkistaa, ettei toinen huijannut.

- Lisäkortteja jaetaan samaan tapaan. Bob valitsee jäljellä olevista sanomista.
- Protokollasta on myös useamman pelaajan versio.
- Ongelmia:
  - monimutkaisuus ja kalleus
  - lopussa molempien kädet paljastuvat. Huijaus pitäisi voida estää ilman, että avaimia paljastetaan.