Tampereen yliopisto

Leevi Yliollitervo

SELAIMEN SORMENJÄLJITYSMENE-TELMIÄ

Informaatioteknologian ja viestinnän tiedekunta
Kandidaattitutkielma
Toukokuu 2023

TIIVISTELMÄ

Leevi Yliollitervo: Selaimen sormenjäljitysmenetelmiä
Kandidaattitutkielma
Tampereen yliopisto
Tietojenkäsittelytieteiden tutkinto-ohjelma
Toukokuu 2023

Selaimen sormenjäljitys on yhä yleisempi keino seurata käyttäjien toimintaa verkossa. Sormenjäljittämisessä kerätään tietoa selaimesta ja sen ympäristöstä, kuten asennetuista fonteista, näytön resoluutiosta ja selaimen lisäosista. Näiden avulla käyttäjästä voidaan luoda yksilöllinen profiili, jota voidaan käyttää käyttäjän seurantaan ja identifiointiin. Sormenjäljitys on herättänyt huolta käyttäjien yksityisyydestä, sillä käyttäjäprofiilin luominen ja seuranta ilman suostumusta, voi johtaa henkilökohtaisen tiedon väärinkäyttöön.

Tämän tutkielma toteutettiin kirjallisuuskatsauksena pohjautuen aikaisempaan tutkimukseen sormenjäljitysmenetelmistä ja niiden toimivuudesta. Sen tavoitteena on selvittää, mitä ja millaisia menetelmiä voidaan käyttää selaimen sormenjäljittämiseen. Ensin työssä käsitellään sormenjäljityksen toimintaa yleisesti. Tämän jälkeen käydään tarkemmin läpi eri sormenjäljitysmenetelmiä. Menetelmistä esitetään niiden toimintaperiaatteita, hyöty- ja haittapuolia, sekä esitetään esimerkkejä käytetymmistä tekniikoista. Lopuksi käsitellään sormenjäljityksen käyttökohteita ja torjuntaa.

Sormenjäljitykseen on olemassa erilaisia menetelmiä. Siihen voidaan käyttää esimerkiksi selaimen ominaisuuksia, JavaScript-moottoria, HTML5-kangasta ja laitteistotehokkuuden mittausta. Näitä menetelmiä voidaan käyttää moniin erilaisiin sormenjäljitys implementaatioihin, joilla on omat hyötynsä ja haittansa. Lisäksi uudet tutkimukset löytävät jatkuvasti uusia menetelmiä. Uudet menetelmät perustuvat usein sivukanavoihin, joten niiden tunnistaminen on perinteisiä menetelmiä hankalampaa.

Vaikka sormenjäljityksellä on hyödyllisiä käyttökohteita, kuten petosten havaitseminen ja verkkosivustojen optimointi, niin sitä voidaan käyttää myös haitallisiin tarkoituksiin, kuten verkkoseurantaan ja kohdennettuun mainontaan. Tämä on herättänyt huolta sormenjäljittämisen vaikutuksista käyttäjän yksityisyyteen ja turvallisuuteen verkossa. Lisäksi sormenjäljittämiseen käytettävät teknologiat kehittyvät nopeasti, joten sen torjumiseen tarvitaan jatkuvaa tutkimusta.

Avainsanat: selaimen sormenjäljitys, verkkoseuranta, tietoturva

SISÄLLYSLUETTELO

1 Johdanto

Käyttäjän seuranta verkossa on entistä yleisempää, verkkosivujen ja mainostajien pyrkiessä seuraamaan käyttäjien käyttäytymistä ja mieltymyksiä. Yksi tärkeimmistä käyttäjän seurantamenetelmistä on selaimen sormenjäljitys.

Selaimen sormenjäljityksessä kerätään eri tekniikoilla tietoa verkkosivua selaavan käyttäjän verkkoselaimesta ja selainympäristöstä. Kerättyä tietoja yhdistämällä voidaan luoda käyttäjälle yksilöllinen tunnus. Vaikka tätä tunnusta voidaan käyttää turvallisuuden tehostamiseen, sen avulla on myös mahdollista seurata käyttäjää eri sessioiden, verkkosivujen tai joissain tapauksissa eri selainten välillä (Yinzhi at al., 2017). Tämä on herättänyt huolta käyttäjien yksityisyydestä verkossa.

Selaimen sormenjäljitys on kasvamassa yhdeksi käytetymmistä käyttäjänseurantamenetelmistä, mutta suuri osa käyttäjistä ei ole tietoisia sen olemassaolosta (Englehardt & Narayanan, 2016)(Iqbal et al., 2021). Viimeisempien tutkimusten mukaan yli 10 % 100 000 suositummasta verkkosivusta sisältää sormenjäljittämiseen käytettävää koodia (Iqbal et al., 2021). Sormenjäljitystä käyttävien sivujen osuus on noin kolminkertaistunut viidessä vuodessa (Englehardt & Narayanan, 2016). Kasvun voidaan katsoa johtuvan evästepohjaisten menetelmien reguloimisesta ja estämisestä. GDPR lakien myötä evästeille on asetettu käyttäjän hyväksyntä vaatimuksia. Lisäksi lakien saaman mediahuomion vuoksi käyttäjät ovat tietoisempia evästeiden olemassaolosta ja tarkoituksesta. Tämän vuoksi useat käyttäjät päättävät evätä evästeiden käytön. Modernit selaimet myös antavat käyttäjälle mahdollisuuden tarkastella ja poistaa käytössä olevia evästeitä. Toisin kuin evästeet, sormenjäljitys mahdollistaa käyttäjän seuraamisen huomaamattomasti, sillä sormenjäljitys ei tallenna pysyvää tietoa käyttäjän päätelaitteelle. Tämä mahdollistaa sen implementoinnin moniin eri verkkoelementteihin, kuten esimerkiksi upotettuihin mainoksiin. Tämän vuoksi on tärkeää kehittää menetelmiä sormenjäljittämisen tunnistamiseen ja estämiseen.

Luettavuuden vuoksi tässä tutkielmassa selaimen sormenjäljityksestä käytetään myös termiä sormenjäljitys. Lisäksi rajatun ajan ja laajuuden vuoksi, tutkielmassa käsitellään ainoastaan selaimen sormenjäljitystä. Mobiiliekosysteemissä toimivat sormenjäljitysmenetelmät eroavat huomattavasti tietokoneissa käytettävistä menetelmistä. Mobiililaitteiden sormenjäljitysmenetelmät riippuvat usein laitteiston ominaisuuksista, joita ei löydy suurimmasta osasta tietokoneista. Tämän vuoksi mobiiliselainten sormenjäljitystä voidaan pitää omana tutkimusalueenaan.

Tunnistamisen ensimmäinen vaihe on ymmärtää miten sormenjäljitysmenetelmät toimivat. Siksi tässä tutkielmassa käsitellään tutkimuskysymystä: mitä ja millaisia menetelmiä voidaan käyttää selaimen sormenjäljittämiseen? Luvussa kaksi käsitellään, millaisilla menetelmillä tutkielma on suoritettu. Kolmannessa luvussa esitetään menetelmiä, jotka ovat yhteistä eri tutkimusten välillä. Lisäksi menetelmiä käydään läpi kategorioittain ja

jokaisesta kategoriasta kerrotaan havainnollistava esimerkki. Neljännessä luvussa kerrotaan sormenjäljityksen käyttökohteista ja esitellään menetelmiä sormenjäljityksen tunnistamiseen ja torjumiseen. Viimeisessä ja viidennessä luvussa esitetään yhteenveto tutkielmasta.

2 Tutkimusmenetelmä

Ainestoa on haettu: Tampereen yliopiston Andor, ACM Digital Library, Google Scholar ja IEEE Electronic Library tietokannoista. Hakusanoina on käytetty: “Browser fingerprinting”, “Cross-browser fingerprinting”, “Browser tracking” ja “Device fingerprinting”. Alustava rajaus suoritettiin valitsemalla enintään 10 vuotta vanhat aineistot. Tästä poikkeuksena ovat lähteet, joita käytetään vertailussa havainnollistamaan sormenjäljityksen kehitystä. Lisäksi lähteen täytyy käsitellä tietokoneiden sormenjäljittämistä, sillä tämä työ ei käsittele mobiililaitteita. Lopuksi lähteen täytyy olla luotettavalta julkaisulta. Lähteen sopivuutta tarkasteltiin lukemalla siitä aluksi tiivistelmä, avainsanat ja väliotsikot.

Lähteet on jaettu kolmeen kategoriaan niiden tarkemman aiheen mukaan. Ensimmäinen kategoria sisältää lähteitä, jotka tutkivat tai pyrkivät parantamaan jotain tiettyä menetelmää. Toisessa kategoriassa ovat lähteet, jotka tutkivat sormenjäljityksen toimintaa suuremmalla tietokannalla. Kolmas kategoria sisältää lähteitä, jotka keskittyvät sormenjäljityksen tunnistamisen tutkimiseen.

3 Tutkimustulos

Modernit sormenjäljityspalvelut yhdistävät useita eri menetelmiä (Iqbal et al., 2021). Mittausmenetelmien määrän kasvaessa, mahdollisten sormenjälkien määrä kasvaa eksponentiaalisesti ja täten on teoriassa mahdollista luoda yksilöllinen tunniste jokaiselle käyttäjälle. Todellisuudessa kaikki arvot, mitä eri sormenjäljitykseen käytettävät menetelmät voiva mitata, eivät ole samanarvoisia. Tämän vuoksi mitattavien arvojen valinta on merkittävä osa sormenjäljittämisen toimintaa. Esimerkiksi IP-osoite mahdollistaa käyttäjän tarkan identifioinnin, mutta osoite saattaa vaihtua (Yinzhi et al., 2017). Vastakohtana on käyttäjän käyttöjärjestelmä, sillä suurin osa käyttäjistä käyttää samaa käyttöjärjestelmää (Gómez-Boix et al., 2018). Eri arvot sisältävät täten erilaisia hyötyjä ja haittoja, joten sormenjäljitystä varten on kehitetty menetelmiä mitattujen arvojen valintaan ja painottamiseen. Arvojen hyödyllisyydestä sormenjäljityksessä käytetään termejä *luotettavuus* (stability) ja *entropia* (entropy).

Luotettavuus mittaa kuinka luotettava mitattu arvo on ajan kuluessa. Luotettavana arvona voidaan pitää esimerkiksi käyttäjän käyttöjärjestelmää, sillä tämän vaihtuminen useasti on epätodennäköistä. Epäluotettavana arvona voidaan pitää esimerkiksi kannettavan tietokoneen akun varausta, sillä sen arvo eri mittauskerroilla on usein satunnainen.

Entropia on termi, joka kuvaa epäjärjestystä tai sattumanvaraisuutta. Termiä käytetään usein fysiikassa ja tietojenkäsittelytieteessä kuvaamaan järjestelmän tai datan ennustamattomuutta. Korkea entropia tarkoittaa suurta epäjärjestystä tai monimuotoisuutta, kun taas matala entropia tarkoittaa vähäistä epäjärjestystä tai yksinkertaisuutta. Korkean entropian arvona voidaan pitää käyttäjän IP-osoitetta, sillä se on uniikki suurimalle osasta käyttäjistä. Matalan entropian arvona voidaan pitää esimerkiksi käyttäjän selainohjelmistoa, kuten Chrome tai Firefox, sillä niitä on vain vähän.

Eri mittausarvoista on hankala päätellä, mitkä arvot parantavat tai heikentävät sormenjälkeä. Esimerkiksi akun varausta voidaan käyttää tunnistamaan, onko kyseessä pöytäkone vai kannettava tietokone (Englehardt & Narayanan 2016). Samoin selaimen malli toimii yksilöllisenä arvona, jos käytössä on harvinaisempi selainmalli. Tämän vuoksi sormenjäljittämisessä käytetään hyväksi informaatiotiedettä, joka auttaa löytämään arvot, jotka tuottavat tarkimman sormenjäljen. Nykyään tutkimus on siirtymässä käyttämään enemmän koneoppimista.

Pääsäännöllisesti uudet menetelmät perustuvat *sivukanavoihin* (side-channel). Sivukanava on tietoturvaan liittyvä käsite, joka tarkoittaa epäsuoraa tapaa saada tietoa tietojärjestelmästä, hyödyntäen sen ominaisuuksia tai heikkouksia. Sivukanavoiden käyttö johtuu siitä, että selainvalmistajat poistavat käytöstä ominaisuuksia, joita on voitu suoraan käyttää sormenjäljittämiseen. Tästä esimerkkinä on tuen lopettaminen Flash-liitännäisille, jotka olivat yksi tärkeimmistä menetelmistä sormenjäljittämisessä. Sivukanavamenetelmät ovat myös hyödyllisiä keräämään tietoa, joka ei olisi normaalisti saatavilla.

3.1 Selaimen ominaisuudet

Jokaisen verkkosivupyynnön yhteydessä selain lähettää tietoa käyttäjästä, jota verkkosivu voi käyttää hyödyksi sivun sisällön mukauttamiseen. Esimerkiksi verkkosivut voivat automaattisesti muuttaa kielen käyttäjän käyttämän kielen mukaisesti. Lisäksi verkkosivujen taustajärjestelmät sisältävät usein monia selainkohtaisia optimointeja, jotka ovat tarkoitettu parantamaan sivuston suorituskykyä ja latausaikoja. Näitä tietoja voidaan käyttää myös sormenjäljittämiseen (Upathilake et al., 2015).

Nykyään verkkosivut sisältävät monipuolista multimediasisältöä, kuten videota, ääntä ja animaatioita. Tämän sisällön esittämiseen on kehitetty useita selainlaajennuksia, kuten *Adobe Flash*, joka mahdollisti interaktiivisen sisällön esittämisen verkkosivuilla. Flash on korvattu nykyään HTML5:llä ja JavaScriptillä, jotka ovat tehokkaampia ja turvallisempia tekniikoita. Näiden teknologioiden rajapinnat tarjoavat myös mahdollisuuksia sormenjäljitettävän tiedon keräämiseen (Upathilake et al., 2015).

Haittapuolena selaimen ominaisuuksien käyttämisessä sormenjäljittämisessä on, että ne eivät luonnostaan toimi eri selainten välillä. Niiden tuottama tieto on kuitenkin helposti saatavilla, joten sitä käytetään usein sormenjäljityksessä.

3.1.1 Kirjasintyypit ja liitännäiset

Jotta verkkosivu voi näyttää käyttäjälle tekstiä haluamallaan kirjasintyypillä eli fontilla, täytyy sen tietää mitä eri fontteja käyttäjän päätelaitteelle on asennettu. Ennen verkkosivut pystyivät hakemaan Flash-rajapinnan avulla listan fonteista, joita käytettiin tekstin näyttämiseen sivulla. Lista fonteista vaihteli eri käyttäjien välillä, joten sitä käytettiin sormenjäljittämiseen (Kaur et al., 2017). Nykyään Flash tuen loputtua, fonttien keräämiseen käytetään tyypillisesti sivukanavoita, joita käsitellään kappaleessa 3.3.

Toinen yleinen sormenjäljittämisen kohde on selaimeen asennetut liitännäiset (Kaur et al., 2017). Liitännäiset ovat selaimen toimintaa tukevia lisäohjelmistoja, jotka mahdollistavat esimerkiksi PDF-tiedostojen selaamisen. Käyttäjien eri tarpeiden mukaan selaimeen on asennettu erilaisia liitännäisiä. Jokaisella liitännäisellä on myös versionumero, joka lisää käyttäjän liitännäislistan yksilöllisyyttä. Asennetut liitännäiset ovat kuitenkin poistumassa käytöstä sormenjäljitysmenetelmänä, sillä yhä enemmän niiden tarjoamista ominaisuuksista on sisällytetty valmiiksi selaimiin (Umar et al., 2021).

3.2 JavaScript

JavaScript on noussut internetin ydinteknologiaksi. Melkein kaikki modernit verkkosivut käyttävät JavaScriptiä. Tätä varten kaikissa moderneissa selaimissa on sisäänrakennettu JavaScript-moottori, joka mahdollistaa ohjelmakoodin ajamisen suoraan käyttäjän omalla laitteella. Lisäksi se tarjoaa useita eri rajapintoja ja ominaisuuksia verkkosivun kehittäjän käytettäväksi, kuten esimerkiksi tietoa käyttäjän audiolaitteista tai akun varauksesta. Monet näistä ominaisuuksista on tarkoitettu mukauttamaan verkkosivu käyttäjän päätelaitteelle sopivaksi, sekä sivun interaktiivisuuden laajentamiseksi. JavaScriptin kautta verkkosivulla on pääsy moniin sormenjäljitettäviin ominaisuuksiin (Kaur et al., 2017)(Upathilake et al., 2015). JavaScriptillä saatavien ominaisuuksien lisäksi selaimen JavaScript implementaatiota voidaan käyttää sormenjäljitykseen (Mulazzani et al., 2012).

JavaScriptin käyttö sormenjäljityksessä on helppoa ja tehokasta. Lisäksi sen käyttäminen sormenjäljityksessä sisältää vain vähän haittoja. Koska JavaScript on integroitu melkein kaikkeen verkkosisältöön, on sen estäminen tehokkaasti hankalaa ja siksi kattavien estomenetelmien toteuttaminen vaatii huomattavia resursseja.

JavaScriptin tarjoamien rajapintojen käyttäminen sormenjäljityksessä on altis virtualisoinnille, sillä sen avulla selaimet voivat eristää JavaScript-sovellukset oikeasta laitteistosta. Tämä poistaa suuren osan JavaScript-rajapintojen soveltuvuudesta sormenjäljitykseen. Lisäksi selaimet voivat vaatia käyttäjältä lupaa rajapintojen suorittamiselle. JavaScriptin käyttö sormenjäljityksessä, on mahdollista tunnistaa seuraamalla ajettavien skriptien funktio kutsuja (Iqbal et al., 2021). Tarkastelemalla skriptin kutsumia funktioita, on mahdollista löytää sormenjäljittämiseen käytettäviä skriptejä.

3.2.1 ECMAScript standardi

ECMAScript on JavaScript-kielen standardi. Siinä määritellään kielen syntaksi, semantiikka ja keskeiset ominaisuudet, joita selainvalmistajat seuraavat omassa JavaScript-moottorin toteutuksessaan. Koska ECMAScript ei määrittele miten se pitää toteuttaa, löytyy eri selainten implementaatioista joitain eroja. *ECMAScript-vaatimustenmukaisuustestit* (ECMAScript conformance tests) ovat joukko testejä, joilla varmistetaan, että JavaScript-moottori on ECMAScript-standardin mukainen. Näitä testejä voidaan käyttää sormenjäljittämisessä analysoimalla testien tuloksia käyttäjän selaimessa (Mulazzani et al., 2012). Tällä voidaan tunnistaa esimerkiksi käytettävän JavaScript-moottorin versio (Mulazzani et al., 2012). ECMAScript-standardin laajuuden vuoksi, sen eri toteutuksissa on pieniä eroja, jotka näkyvät erilaisina tuloksina vaatimustenmukaisuustesteissä. Testien tulokset voivat vaihdella monista syistä, kuten esimerkiksi puuttuvista ominaisuuksista tai virheellisistä toteutuksista.

Lisäksi vaatimustenmukaisuustestien tulokset voivat paljastaa tietoa käyttäjän laitteistosta ja käyttöjärjestelmästä. Testit voivat esimerkiksi edellyttää tiettyjen laitteisto-ominaisuuksien, kuten SIMD-käskyjen käyttöä tai käyttäytyä eri tavoin eri käyttöjärjestelmissä (Saito et al, 2016).

3.3 HTML5 kangas

Moderneista menetelmistä tärkeimpänä voidaan pitää HTML5:n tarjoamaa *kangaselementtiä* (canvas), joka on tarkoitettu 2D-grafiikan renderöintiin (Englehardt & Narayanan 2016). Kangaselementtiä pidetään tehokkaimpana sormenjäljitysmenetelmänä, sillä sen avulla voidaan nopeasti ja luotettavasti testata useita eri käyttäjän päätelaitteen ominaisuuksia. Kankaalta kerätyt pikselit ajetaan hajautusfunktion läpi ja sen palauttamaa arvoa voidaan käyttää suoraan sormenjälkenä. Sormenjälki on mahdollista rakentaa niin, että pikselien arvot eivät riipu käytetystä selaimesta (Cao et al., 2017). Kangasta käytetään paljon normaaleissa nettisivun toiminnoissa, joten sen käyttöä sormenjäljityksessä on erityisen hankala tunnistaa ja estää (Iqbal et al., 2021). Tor-selain oletuksena estää verkkosivulta kankaan pikselien arvojen lukemisen, mutta useimmissa tapauksissa tämä hajottaa verkkosivun toimintoja.

Kangas-sormenjäljentämiseen liittyy joitakin teknisiä seikkoja, jotka voivat rajoittaa sen tehokkuutta. Yksi tekninen rajoitus on selaintuki, sillä kangas-sormenjäljityksen käyttämä kangaselementti ei ole tuettu kaikissa selaimissa. Kuitenkin modernit selaimet tukevat kangaselementin käyttöä, joten tämä ongelma on vähitellen poistumassa. Yhteensopivuusongelmat eri selainversioiden, käyttöjärjestelmien ja laitteiden välillä voivat myös vaikuttaa tekniikan tarkkuuteen ja luotettavuuteen (Yinzhi et al., 2017). Koska kangas-sormenjälki on luonnostaan riippuvainen kangaselementistä, muutokset kangaselementin toteutukseen voivat vaatia muutoksia sormenjäljityksen toteutuksessa. Myös muutokset

laitteistorajapintoihin voivat vaikuttaa sormenjälkeen. Tämän vuoksi kangassormenjälki vaatii jatkuvaa ylläpitoa. Viimeisenä kangas-sormenjäljen käyttö voi vaikuttaa haitallisesti käyttäjäkokemukseen lisäämällä käsittelyaikaa ja aiheuttamalla viiveitä verkkosivujen lataamisessa.

3.3.1 WebGL

WebGL-rajapinta mahdollistaa 2D- ja 3D-renderöinnin HTML5-kankaalle. Vastaavasti se pohjautuu OpenGL:ään. OpenGL on avoin standardi, joka jättää implementaation grafiikkapiirien valmistajien vastuulle. Tämä vuoksi eri valmistajien grafiikkapiirit saattavat käyttää hieman erilaisia algoritmeja ja toimintatapoja sen toteuttamiseen. Tämä johtaa pieniin eroavaisuuksiin kuvan piirtämisessä eri laitteilla, jota voidaan käyttää hyväksi sormenjäljittämisessä (Yinzhi et al., 2017). Käytännössä nämä erot ovat liian pieniä havaittavaksi normaalissa käytössä, mutta pieniäkin muutoksia on mahdollista käyttää sormenjäljityksessä. Yleensä sormenjäljitystoteutuksissa kangas renderöidään piilotettuna.

Vaikka renderöitävä grafiikka voi olla mitä tahansa, voidaan paremmalla toteutuksella luoda tarkempi ja luotettavampi sormenjälki. Renderöitävään grafiikkaan vaikuttavat selaimen ja laitteiston ominaisuudet. Renderöimällä tarkkaan valittuja yksinkertaisia kappaleita, on mahdollista eristää tiettyjä renderöinti ominaisuuksia, kuten *antialiasointi* (antialiazing) tai *läpinäkyvyys* (transparency). Valitsemalla mitattavat ominaisuudet huolellisesti on mahdollista esimerkiksi muodostaa sormenjälki, joka ei ole riippuvainen käyttäjän ikkunan koosta tai selaimesta. Tämä kasvattaa sormenjäljen luotettavuutta. (Cao et al., 2017)

3.3.2 Fontti sivukanava

Flash-liitännäisen kautta oli mahdollista saada fonttilista, mutta Flashin tuen päätyttyä tämä menetelmä vanhentui. Tämän vuoksi on kehitetty sivukanavaan perustuva menetelmä, joka mahdollistaa fonttien keräämisen. Sivukanava perustuu eri fonttien pikselikoon vaihteluun. Menetelmässä yritetään renderöidä tekstiä jollain fontilla ja vertaamalla tekstin kokoa pikseleissä ennalta mitattuun arvoon. Jos fonttien koot täsmäävät voidaan todeta, että testattu fontti löytyy käyttäjän laitteelta. Valitsemalla huolellisesti testattavat fontit, voidaan muodostaa selaimesta riippumaton sormenjälki, kuten WebGL:ään perustuvissa menetelmissä (Cao et al., 2017).

3.4 Tehokkuuden mittaus

Tietokoneiden laitteisto, kuten keskusprosessori ja näytönohjain, vaihtelevat merkittävästi eri käyttäjien välillä. Laitteistovalmistajat julkaisevat jatkuvasti uusia komponentteja, minkä vuoksi eri käyttäjät päätyvät usein erilaisista osista koostuviin tietokoneisiin. Laitteiston tuottama sormenjälki toimii myös eri selainten välillä (Cao et al., 2017). Tämän vuoksi ne toimivat erinomaisena sormenjäljittämisen kohteena. Suurin osa käyttäjän

laitteistotiedoista ei ole kuitenkaan suoraan saatavilla, joten niiden keräämiseen tarvitaan sivukanavamenetelmiä. Yksi sivukanatyyppi on *tehokkuuden mittaus* (benchmarking). Mittaamalla eri prosessien välisiä ajoaikoja voidaan luoda käyttäjälle sormenjälki. Prosessi voi olla esimerkiksi jokin algoritmi, kuten md5-kryptografia-algoritmi tai JavaScript funktio, kuten Regex. Valitsemalla eri testattavia prosesseja, voidaan testata käyttäjän laitteen eri ominaisuuksia, kuten keskusprosessorin ytimien määrä tai SIMD-käskyjen olemassaoloa.

Haittapuolena tehokkuuden mittauksessa on, että se vaatii merkittävän määrän resursseja käyttäjän laitteelta. Lisäksi riippuen testistä, tehokkuuden mittaus saattaa kestää useita sekunteja. Tehokkuuden mittaukseen vaikuttavat myös monet tekijät, jotka eivät ole mittaajan hallittavissa, kuten käyttöjärjestelmän *vuoronnus* (scheduling), muut samanaikaiset prosessit ja tehon rajoitus (Mowery et al., 2012). Tekemällä useita ja pidempiä mittauksia on mahdollista kasvattaa tuloksen luotettavuutta testaus ajan kustannuksella.

3.4 Keskusprosessorin sormenjäljitys

Moderneissa prosessoreissa on ominaisuuksia, jotka on tarkoitettu nopeuttamaan tietyn prosessin laskentaa. Riippuen prosessorin mallista ja valmistajasta, näiden ominaisuuksien implementaatio ja olemassaolo voivat vaihdella. Tutkimukset ovat osoittaneet, että mittaamalla eri prosessien ajoaikoja voidaan tunnistaa näiden ominaisuuksien olomassaoloa (Saito et al., 2016).

Ajoaikojen mittaaminen eri prosesseille voi toimia sivukanavana selaimen version ja käyttöjärjestelmän tunnistamiseen. Uudemmissa selainversioissa tapahtuu usein optimointeja, jotka vaikuttavat eri prosessien ajoaikoihin. Esimerkiksi Chrome 2.0:ssa md5-algoritmin ja Regex-funktion ajoaikojen normalisoitu ero on 100 %. Chrome 3.0:ssa Regex-laskenta nopeutuu 5 %, joten ajoaikojen ero on nyt 105 % (esimerkkiluvut eivät vastaa todellisia aikoja). Vertaamalla aikaeroja ennalta mitattuihin arvoihin voidaan päätellä, selaimen versio, käyttöjärjestelmä ja prosessorin arkkitehtuuri. Todellisuudessa implementaatiossa olisi testattava useita prosesseja, koska kahden prosessin väliset erot eivät ole näin selkeitä ja sisältävät merkittävää hajontaa. (Mowery et al., 2012)

4 Keskustelu

Selaimen sormenjäljittämistä voidaan käyttää, sekä hyödyllisiin, että haitallisiin tarkoituksiin. Koska sormenjäljittäminen mahdollistaa käyttäjän seurannan verkossa, jakaa se monia käyttötarkoituksia muiden seurantamenetelmien, kuten evästeiden kanssa. Monien muiden seurantamenetelmien tavoin verkkosivujen välinen seuranta edellyttää niiden välisen yhteisen seurauspalveluntarjoajan. Tämän vuoksi pelkällä sormenjäljityksellä on vain vähän haitallisia käyttökohteita.

Sormenjäljityksen mahdollistamaa seurantaa käytetään yleisimmin kohdennetun sisällön tarjonassa. Sormenjäljityksellä voidaan seurata käyttäjää eri sessioiden ja verkkosivujen välillä, muodostaen kuvan käyttäjän mieltymyksistä. Uutissivusto voi esimerkiksi tarjota käyttäjälle relevanttia sisältöä selaushistorian avulla. Lisäksi sivusto voi näyttää uutisia, jotka vastaavat käyttäjän aikaisemmin lukemia uutisia. Verkkosivujen välinen seuranta tapahtuu seurantapalvelujen avulla, jotka keräävät sormenjälkiä usealta eri sivustolta. Seurantaa käytetään myös kohdennettujen mainosten tarjoamiseen.

Sormenjäljityksellä on myös hyödyllisiä käyttökohteita. Sitä voidaan käyttää apuna käyttäjän identiteetin varmistamisessa. Jos käyttäjän sormenjälki muuttuu huomattavasti, voidaan esimerkiksi vaatia käyttäjää syöttämään sähköpostin tai tekstiviestin kautta lähetettävä vahvistuskoodi. Sormenjäljitystä voidaan myös käyttää *istunnon turvaamiseen* (session security), vaatimalla käyttäjää kirjautumaan uudelleen, jos sormenjälki muuttuu istunnon aikana (Unger et al., 2013). Tällä voidaan estää väliintulohyökkäykset, joissa hyökkääjä on saanut haltuunsa käyttäjän istuntotunnuksen.

Viimeisenä sormenjäljitystä voidaan käyttää kohdennettuihin kyberhyökkäyksiin (Upathilake et al., 2015). Tunnistamalla uhrin laiteominaisuuksia, kuten käyttöjärjestelmä tai selainversio, voi haittaohjelma varmistaa, että sen käyttämä heikkous löytyy käyttäjän laitteelta.

Selaimen sormenjäljittämiseen käytetään monia menetelmiä ja tämän vuoksi sen estämiseksi on monia eri lähestymistapoja. Koska monet sormenjäljittämismenetelmistä hyödyntävät ominaisuuksia, jotka ovat välttämättömiä moderneille verkkosivuille, on niiden tunnistaminen ja estäminen hankalaa.

Yksi ratkaisu sormenjäljittämisen estämiseen on standardisointi (Laperdrix et al., 2015). Jos jokaisen käyttäjän päätelaite näyttää verkkosivulle samalta, ei ole mahdollista luoda yksilöllistä sormenjälkeä. Esimerkiksi selaimille voitaisiin määritellä sallittu standardisoitu fonttilista, jolloin verkkosivulla ei olisi tarvetta tarkistaa saatavilla olevia fontteja. Fonttilistan standardisointi on jo aloitettu ja selaimilla on käytössä WebFont standardin mukainen lista verkossa käytettävistä fonteista. Kuitenkin kuten luvussa 3.3 huomattiin, verkkosivulla on yhä pääsy käyttäjän fontteihin sivukanavan kautta. Käyttäjäagenttista ja liitännäislistasta voitaisiin myös tehdä vähemmän verbaalinen. Tarkan versionumeron sijaan ilmoitettaisiin geneerinen versionumero (Chrome 110 vrt. Chrome 110.0.0.1).

Standardisoinnin lisäksi on ehdotettu selaimen tai selainympäristön virtualisointia (Laperdrix et al., 2015). Äärilaitana on selaimen ajaminen virtuaalikoneessa, jolloin sormenjälki olisi aina sama riippumatta päätelaitteen todellisesta laitteistosta. Virtuaalikoneen ajaminen on kuitenkin tehokkuuden kannalta raskasta. Toisena ääripäänä on ominaisuuksien virtualisointi selaimessa. Tämä on huomattavasti kevyempää, mutta jättää

mahdollisuuksia sormenjäljitykselle, jos kaikkia sormenjäljitettäviä ominaisuuksia ei virtualisoida (Cao et al., 2017).

Vastakohtana standardisoinnille, voidaan myös satunnaistamista käyttää puolustautumiseen sormenjäljitystä vastaan. Jos sormenjäljityksen mittaamat arvot vaihtelevat jokaisella mittauskerralla, on mahdotonta luoda historiaa arvoista, jolla voitaisiin seurata käyttäjää. Monia arvoja voidaan satunnaistaa, kuten käyttäjäagentti, kangas, fontit tai näytön koko. On kuitenkin tärkeää huomata, että satunnaistetut kokoonpanot voivat mahdollisesti rikkoa normaaleja toimintoja, jotka riippuvat näistä arvoista.

Kangaspohjaisten sormenjäljitysmenetelmien estäminen on hankalaa johtuen sen laajasta käytöstä verkkosivuissa. Jäljittämisen estämiseksi on ehdotettu satunnaisarvojen lisäämistä kankaaseen verkkosivun kerätessä pikselit kangaselementistä. Tutkimukset kuitenkin osoittavat, että tämä menetelmä ei ole riittävä sormenjäljittämisen estämiseksi (Iqbal et al., 2021). Joissain tapauksissa näitä lisättyjä satunnaisarvoja voidaan myös käyttää sormenjäljittämiseen (Iqbal et al., 2021).

Uudeksi menetelmäksi on noussut koneoppimisen ja tekoälyn käyttäminen, sekä sormenjäljittämisen tunnistamisessa ja estämisessä (Bird et al., 2020). Opettamalla tekoälylle sormenjäljittämisessä käytettäviä avainsanoja ja ohjelmakoodia, on sen avulla mahdollista tunnistaa sormenjäljitystä (Iqbal et al., 2021). Tekoälyä voidaan käyttää myös tunnistamaan ja estämään sormenjäljityspalveluita verkkoliikenteen perusteella (Iqbal et al., 2021).

Viimeisenä menetelmänä on sormenjäljittämiseen käytettävien teknologioiden estäminen. Rajoittamalla oletuksena verkkosivua ajamasta mitään JavaScript-koodia tai pääsyä HTML5-kangaselementtiin, voidaan estää merkittävä osa sormenjäljittämisestä. Tämä kuitenkin melkein aina rikkoo verkkosivun toimintoja tai estää pääsyn verkkosivulle kokonaan (Englehardt & Narayanan 2016) (Iqbal et al., 2021).

5 Yhteenveto

Selaimen sormenjäljittämismenetelmiä käytetään seuraamaan käyttäjien verkkokäyttäytymistä. Näihin tekniikoihin kuuluvat selaimen eri ominaisuuksien, kuten asennettujen fonttien, näytön resoluution ja selainliitännäisten kerääminen, jonka avulla voidaan luoda kullekin käyttäjälle yksilöllinen profiili. Vaikka sormenjäljityksellä on monia hyödyllisiä käyttötarkoituksia, kuten petosten havaitseminen ja verkkosivustojen optimointi, niin voidaan sitä käyttää myös haitallisiin tarkoituksiin, kuten verkkoseurantaan ja kohdennettuun mainontaan. Tämä on herännyt huolta sormenjäljityksen vaikutuksista yksityisyyteen ja turvallisuuteen verkossa.

Selaimen sormenjälkitekniikoiden laajamittainen käyttöönotto on seurausta perinteisten seurantamenetelmien, kuten evästeiden rajoituksista. Monissa seurantamenetelmissä käyttäjät voivat helposti poistaa tai estää niiden käytön. Selaimen sormenjäljitys on sen

sijaan vaikeampi estää, sillä se ei perustu pysyvän tiedon tallentamiseen käyttäjän päätelaitteelle. Koska monet käyttäjät eivät ole tietoisia sormenjäljittämisestä, sitä käytetään usein heidän tietämättään tai suostumuksettaan.

Tässä tutkielmassa on esitetty sormenjäljittämiseen käytettäviä tekniikoita, kuten kangas- ja keskusprosessorin sormenjäljitys. Tutkimuksissa löydetään jatkuvasti uusia menetelmiä, jotka osoittavat kuinka mitä tahansa tietoa voidaan käyttää hyväksi sormenjäljittämisessä. Tutkimukset ovat esimerkiksi osoittaneet kuinka CSS:t (Cascading Style Sheets) voidaan käyttää hyväksi sormenjäljittämisessä (Takei et al., 2015). Tämä on merkittävää, sillä sormenjäljitystä estäessä ei voida olettaa, että jotain teknologiaa ei voida käyttää siihen.

On selvää, että sormenjäljitys aiheuttaa merkittäviä riskejä käyttäjien yksityisyydelle ja turvallisuudelle. Tämän vuoksi tarvitaan lisätutkimusta ja tehokkaampien yksityisyyttä tehostavien tekniikoiden kehittämistä käyttäjien suojaamiseksi. Yksi lupaava lähestymistapa on esimerkiksi virtualisointi- tai hiekkalaatikkotekniikoiden käyttö selaimen eristämiseksi. Lisäksi poliittisten päättäjien on pohdittava sormenjäljittämisen vaikutuksia yksityisyyteen ja turvallisuuteen, sekä käyttöön otettava asianmukaisia säännöksiä käyttäjien oikeuksien suojelemiseksi. Tällaisia säännöksiä voisivat olla esimerkiksi säännökset, joissa edellytetään, että verkkosivustojen on saatava käyttäjältä nimenomainen suostumus ennen selaintietojen keräämistä, tai jotka rajoittavat selaimen sormenjälkien käyttöä verkkoseurannassa ja kohdennetussa mainonnassa.

Verkkosivujen yksityisyyteen liittyvien huolenaiheiden lisääntyessä, on todennäköistä, että sormenjäljittäminen joutuu tulevaisuudessa entistä tarkempaan tarkasteluun ja sääntelyyn. Jossain selaimissa on otettu käyttöön sormenjäljitystä estäviä ominaisuuksia, kuten Firefoxin *Enhanced Tracking Protection* ja Safarin *Intelligent Tracking Prevention*. Lisäksi Euroopan unionin yleisen tietosuoja-asetuksen (GDPR) ja Kalifornian kuluttajansuojalain (CCPA) kaltaiset yksityisyyttä koskevat lait, ovat asettaneet rajoituksia henkilötietojen keräämiselle ja käytölle. On mahdollista, että yksityisyyttä parantavat teknologiat ja säädökset yleistyvät tulevaisuudessa, mikä johtaa lopulta selaimen sormenjälkien käytön vähenemiseen.

Kaiken kaikkiaan sormenjäljityksen tulevaisuus näyttää epävarmalta, sillä vaikka sormenjäljityksen estämiseksi on kehitetty erilaisia vastatoimia, niin sormenjäljittämisteknologia kehittyy nopeasti. Sormenjäljityksen käytön yleistyminen korostaa tarvetta kokonaisvaltaisempaan lähestymistapaan yksityisyyden suojaamiseen verkossa, jossa otetaan huomioon yksittäisten tietojen lisäksi koko tiedonkeruun ja -käytön ekosysteemi.

Lähdeluettelo

Bird, S., Mishra, V., Englehardt, S., Willoughby, R., Zeber, D., Rudametkin, W., & Lopatka, M. (2020). Actions speak louder than words: Semi-supervised learning for browser fingerprinting detection. https://doi.org/10.48550/ARXIV.2003.04463

Englehardt, S., & Narayanan, A. (2016). Online Tracking: A 1-million-site Measurement and Analysis. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* 1388–1401. https://doi.org/10.1145/2976749.2978313

Gómez-Boix, A., Laperdrix, P., & Baudry, B. (2018). Hiding in the crowd: An analysis of the effectiveness of browser fingerprinting at large scale. *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18.* 309–318. https://doi.org/10.1145/3178876.3186097

Iqbal, U., Englehardt, S., & Shafiq, Z. (2021). Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors. *2021 IEEE Symposium on Security and Privacy (SP).* 1143–1161. https://doi.org/10.1109/SP40001.2021.00017

Kaur, N., Azam, S., Kannoorpatti, K., Yeo, C. & Shanmugam B. Browser Fingerprinting as user tracking technology. (2017). *2017 11th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 2017,* 103–111, https://doi.org/10.1109/ISCO.2017.7855963

Laperdrix, P., Rudametkin, W. & Baudry, B. (2015). Mitigating Browser Fingerprint Tracking: Multi-level Reconfiguration and Diversification. *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems,* 98–108. https://doi.org/10.1109/SEAMS.2015.18

Mowery, K., Bogenreif, D., Yilek, S. & Shacham, H. (2012). Fingerprinting Information in JavaScript Implementations. *Proceedings of W2SP 2011. IEEE Computer Society, May 2011.*

Mulazzani, M., Schrittwieser, S., Reschl, P., Leithner, M., Weippl, E. & Huber, M. (2013). Fast and Reliable Browser Identification with JavaScript Engine Fingerprinting. *Web 2.0 Security & Privacy 2013, San Francisco.* https://doi.org/20.500.12708/85714

Saito, T., Yasuda, K., Ishikawa, T., Hosoi, R., Takahashi, K., Chen, Y. & Zalasiński, M. (2016). Estimating CPU Features by Browser Fingerprinting. *2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS),* 587–592. https://doi.org/10.1109/IMIS.2016.108

Takei, N., Saito, T., Takasu, K. & Yamada, T. (2015). Web Browser Fingerprinting Using Only Cascading Style Sheets. *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA),* 57–63. https://doi.org/10.1109/BWCCA.2015.105

Unger, T., Mulazzani, M., Frühwirt, D., Huber, M., Schrittwieser, S. & Weippl, E. (2013). SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting. *2013 International Conference on Availability, Reliability and Security,* 255–261. https://doi.org/10.1109/ARES.2013.33

Upathilake, R., Li, Y. & Matrawy, A. (2015). A classification of web browser fingerprinting techniques. *2015 7th International Conference on New Technologies, Mobility and Security (NTMS),* 1–5. https://doi.org/10.1109/NTMS.2015.7266460

Yinzhi, C., Song, L. & Erik, W. (2017). (Cross-)Browser Fingerprinting via OS and Hardware Level Features. *NDSS Symposium 2017.* https://doi.org/10.14722/ndss.2017.23152