Vakbijlage

Forensisch gebruik van bestandskenmerken en bijbehorende hashalgoritmen

Inhoudsopgave

1. De vakbijlage algemeen

Het Nederlands Forensisch Instituut (NFI) kent een groot aantal typen onderzoeken. Normaal gesproken gaat elk onderzoeksrapport van het NFI vergezeld van een vakbijlage. Deze dient als toelichting op het onderzoek en heeft een zuiver informatief karakter. Het bevat geen zaak-specifieke informatie. Aan het einde van deze vakbijlage zijn een verklarende woordenlijst en een overzicht van bron- en literatuurverwijzingen opgenomen.

2. Inleiding

Diverse teams[1] van de Divisie Digitale en Biometrische Sporen (DBS) van het NFI ontvangen en onderzoeken voornamelijk digitaal materiaal. Ook de resultaten die deze teams opleveren bestaan vooral uit digitale gegevens. Om de integriteit van de inhoud van digitaal materiaal te waarborgen gebruikt DBS *bestandskenmerken*. Bestandskenmerken kunnen hiernaast ook gebruikt worden voor classificatie en identificatie van bestanden.

[1] In ieder geval de teams Forensische Digitale Technologie, Forensische Big Data Analyse en de groep Beeld van het team Forensische Biometrie. Waar verder DBS in deze vakbijlage staat worden deze teams bedoeld.

Bestandskenmerken over de inhoud van digitaal materiaal worden uitgerekend met programma's die gebruik maken van specifieke hiervoor geschikte rekenkundige methoden, *hashalgoritmen* genaamd.

De (Engelse) termen *hash*, *hash value* en *hashwaarde* zijn veelgebruikte synoniemen voor de term bestandskenmerk. Hiernaast wordt een bestandskenmerk ook wel een 'digitale vingerafdruk'[2] van een bestand genoemd.

In deskundigenrapporten van DBS worden vaak bestandskenmerken van het ontvangen en opgeleverde digitale materiaal vermeld. Deze vakbijlage gaat in op bestandskenmerken en legt de belangrijkste forensische toepassingen ervan uit. Ook het gebruik van bestandskenmerken in de praktijk komt aan de orde. Als laatste wordt beschreven welke technische keuzes DBS heeft gemaakt bij de gebruikte hashalgoritmen voor het berekenen van bestandskenmerken.

3. Bestandskenmerken

Een bestandskenmerk is een compacte representatie van de *binaire* (gedigitaliseerde) inhoud van digitaal materiaal, maar verschaft verder geen informatie de door een persoon *interpreteerbare*[3] inhoud van het materiaal. Waar bijvoorbeeld voor een persoon de inhoud van twee afbeeldingen gelijk zijn (dezelfde interpreteerbare inhoud bevatten) kunnen de bestanden binair gezien verschillend zijn (bijvoorbeeld wanneer het opslagformaat van de afbeeldingen verschilt). Waar in deze bijlage over de inhoud van een bestand wordt gesproken word de binaire inhoud bedoeld, tenzij anders vermeld.

Een digitaal bestand bestaat binair gezien uit een reeks nullen en enen (*bits*). Het aantal en de volgorde van deze nullen en enen bepalen de inhoud van het bestand en zijn daarmee kenmerkend voor dat bestand. Door gebruik te maken van een hashalgoritme worden de specifieke nullen en enen van een bepaald bestand omgezet in een veel eenvoudigere en compactere notatie. Een hashalgoritme is een eindige (en meestal complexe) reeks (mathematische) instructies waarmee op basis van de inhoud van het oorspronkelijke bestand een reeks nullen en enen van vaste lengte wordt gegenereerd. Deze gegeneerde reeks is kenmerkend voor de inhoud van het oorspronkelijke bestand en wordt dan ook aangeduid als het "bestandskenmerk" van het oorspronkelijke bestand. Hoewel in deze vakbijlage vooral over (digitale) bestanden wordt gesproken geldt het (forensisch) gebruik van bestandskenmerken voor al het digitale materiaal, dus bijvoorbeeld ook voor de volledige inhoud (één-op-één-kopie, veiliggestelde gegevens, Engels: *image*) van een digitale gegevensdrager.

Zoals eerder aangegeven moet men er zich van bewust zijn dat twee bestanden er voor een gebruiker hetzelfde uit kunnen zien, maar wanner de bestanden op bitniveau verschillend zijn deze met extreem grote waarschijnlijkheid andere bestandskenmerken zullen hebben.

Onder meer om de leesbaarheid te vergroten worden in rapporten van DBS bestandskenmerken niet weergegeven in enen en nullen, maar wordt daarbij de zogenaamde zestientallige (hexadecimale) notatie gebruikt. In die notatie worden vier opeenvolgende nullen en enen genoteerd als één van de volgende zestien tekens: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f (zie Tabel 1). Er wordt hierbij geen onderscheid gemaakt tussen hoofdletters en kleine letters.

Bits	Hexadecimaal	Bits	Hexadecimaal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	a
0011	3	1011	b
0100	4	1100	c
0101	5	1101	d
0110	6	1110	e
0111	7	1111	F

Tabel 1: van bits naar hexadecimale notatie

[2] Deze vergelijking geldt alleen voor het overeenkomstige algemene gevoel van onderscheidbaarheid van een vingerafdruk dan wel bestandskenmerk. Strikt genomen loopt deze vergelijking mank: bijna gelijke bestandskenmerken horen (extreem waarschijnlijk) bij bestanden met compleet verschillende inhoud en zijn dan ook zeer onderscheidend te noemen. Bijna gelijke (*matchende*) vingerafdrukken zijn juist niet onderscheidend.

[3] Voorbeelden van door een persoon interpreteerbare bestanden zijn digitale tekstdocumenten en afbeeldingen (visueel) en digitale audiobestanden (auditief).

Om de leesbaarheid nog verder te vergroten wordt het bestandskenmerk vaak in groepjes van vier tekens, elk teken bevat dus 4 bits, weergegeven. Een voorbeeld van een bestandskenmerk bestaande uit 256 bits, dus 64 tekens, ziet er in deze notatie als volgt uit:

```
2584 878d 8694 6001 3fab 045c 4de2 03d1
ca8a bb44 3b6b 2169 c9a1 4021 8d2b 21bb
```

Elke twee tekens samen, dat wil zeggen 8 bits, wordt een byte genoemd. De lengte van een bestandskenmerk wordt over het algemeen aangegeven in aantallen bytes. De hier genoteerde 64 tekens komen dan ook overeen met 32 bytes.

4. Niet-omkeerbare hashalgoritmen

Bij het gebruik van (goede) hashalgoritmen leidt een wijziging van slechts één bit in een bestand met extreem grote waarschijnlijkheid tot een compleet ander bestandskenmerk dan dat van het bestand voor de wijziging. Dit wordt ook wel het 'watervaleffect' (Engels: *avalanche effect*) genoemd. Zie ter illustratie Tabel 2: de bestandskenmerken (in dit voorbeeld MD5, zie ook paragraaf 7) berekend over bijna gelijke bitreeksen zijn totaal verschillend.

Bitreeks	MD5-bestandskenmerk
1000000000000000	fc60 9b43 cb85 95a9 a832 cbc2 591e d83a
1000000000000001	9d26 f82a f654 8210 454c 017a 3179 c0ec
1000000010000000	688e c893 e933 aff7 2480 5d61 4e43 f68e

Tabel 2: MD5-bestandskenmerken van bijna gelijke bitreeksen

Deze vakbijlage beperkt zich tot een specifieke categorie hashalgoritmen en bijbehorende bestandskenmerken, de zogeheten *cryptografische* of *niet-omkeerbare* hashalgoritmen. In het Engels heten deze *one-way hash algorithms*. Waar dit document de term hashalgoritmen vermeldt, gaat het steeds om dergelijke niet-omkeerbare hashalgoritmen. Ze hebben, naast het genoemde watervaleffect, minimaal de volgende eigenschappen:

1. Één-richting (Engels: *one-way*, *preimage resistant*): het is praktisch gezien onmogelijk[4] om bij een gegeven bestandskenmerk een bestand te vinden/creëren dat bij dat bestandskenmerk hoort.
2. Doel-botsing-bestendig (Engels: *target collision resistant*, *second preimage resistant*): het is praktisch gezien onmogelijk een bestand met andere inhoud te vinden/creëren waarbij het bestandskenmerk identiek is aan het bestandskenmerk van een vooraf gegeven bestand.
3. Botsing-bestendig (Engels: *random collision resistant*): het is praktisch gezien onmogelijk om twee bestanden te vinden met andere inhoud maar met hetzelfde bestandskenmerk.

De beschreven eigenschappen van een nietomkeerbaar hashalgoritme zijn aannames die werkbaar zijn tot het tegendeel is aangetoond. Er is momenteel zelfs geen wiskundig bewijs bekend voor het bestaan van niet-omkeerbare hashalgoritmen. Een nietomkeerbaar hashalgoritme verliest zijn veronderstelde niet-omkeerbare status zodra deze *gekraakt* is. Het kraken van een hashalgoritme is meestal een zoektocht naar bestanden die hetzelfde bestandskenmerk hebben. Vervolgens wordt een methode gezocht om efficiënt dergelijke bestanden te vinden. Zodra een dergelijke methode gevonden is voor een hashalgoritme, dan zijn de beschreven eigenschappen (deels) niet meer van toepassing op dit algoritme. Als een gangbaar hashalgoritme gekraakt is of dreigt te worden, zal dit doorgaans plaatsmaken voor een nieuw hashalgoritme dat niet op dezelfde manier valt te kraken. Het overstappen op een nieuw hashalgoritme is daarom na verloop van tijd mogelijk noodzakelijk. In paragraaf 7.1 wordt de huidige status weergegeven voor elk van de door DBS gebruikt hashalgoritme.

[4] 'Praktisch onmogelijk' kan hier worden gelezen als: 'Zelfs wanneer alle computerkracht van de wereld tegelijk gebruikt zou kunnen worden, is het nog steeds onmogelijk'.

5. Forensische toepassingen

Zolang de eigenschappen *één-richting* en *doel-botsingbestendig* van een niet-omkeerbaar hashalgoritme niet gekraakt zijn, zijn de bestandskenmerken die hiermee uitgerekend zijn forensisch toepasbaar voor:

1. integriteitscontrole van digitaal materiaal (zie paragraaf 5.1);
2. efficiënte identificatie en classificatie van bestanden (zie paragraaf 5.2).

Het maakt voor deze forensische toepassingen niet uit of de eigenschap *botsing-bestendig* van een hashalgoritme al dan niet gekraakt is. Bij de twee andere eigenschappen is een van de twee bestandskenmerken gegeven, hetzij als bestandskenmerk, hetzij als het digitale materiaal waar het bestandskenmerk over berekend moet worden. Hierbij moet dan (ander) digitaal materiaal worden gevonden met hetzelfde bestandskenmerk. Bij *botsingbestendig* is geen bestandskenmerk vooraf gegeven: het is voldoende om twee verschillende stukken digitaal materiaal te vinden of te maken, waarbij het bestandskenmerk gelijk is. Dit laatste komt bij de forensische toepassingen niet voor, daar altijd ontvangen of verzonden digitaal materiaal als bronmateriaal dient, waarbij het berekende bestandskenmerk daarover vast ligt.

5.1. Integriteitcontrole

Integriteitscontrole van digitaal materiaal heeft als belangrijkste doel toevallige wijzigingen in (kopieën van) digitaal materiaal te detecteren. Daarnaast kunnen met integriteitscontrole sommige vormen van bewuste manipulatie van digitaal materiaal gedetecteerd worden.

Het is eenvoudig om, al dan niet opzettelijk, digitaal materiaal te wijzigen. Via bestandskenmerken kunnen verschillende personen eenvoudig aan elkaar laten weten met welk materiaal zij hebben gewerkt. Ook kunnen ze ermee vaststellen of ze met hetzelfde materiaal werken als een ander persoon. De ene persoon, bijvoorbeeld een digitaal rechercheur, rapporteert het bestandskenmerk van het digitale materiaal aan een ander persoon, bijvoorbeeld een onderzoeker van DBS. Die berekent vervolgens het bestandskenmerk opnieuw over (een kopie van) het aangeleverde materiaal. De uitkomst wordt vergeleken met het gerapporteerde bestandskenmerk. Als de uitkomst niet exact gelijk is aan het eerder gerapporteerde bestandskenmerk, dan is het materiaal in de tussentijd gewijzigd. Hoe of waar de gegevens verschillen van de originele gegevens maakt het bestandskenmerk overigens niet duidelijk. Als het resulterende bestandskenmerk gelijk is aan het gerapporteerde bestandskenmerk, dan is het extreem waarschijnlijk dat het materiaal niet gewijzigd is sinds het gerapporteerde bestandskenmerk werd berekend.

5.2. Bestandsidentificatie en –classificatie

Voor identificatie en classificatie van bestanden is het gebruik van bestandskenmerken over het algemeen veel efficiënter omdat bestandskenmerken van een bestand relatief klein zijn. Tegenwoordig zijn bestanden zelf vaak (zeer) groot (enkele gigabytes, een gigabyte is ongeveer 1 miljard bytes). De momenteel door DBS gebruikte maximale lengte van een bestandskenmerk is slechts 32 bytes (64 tekens). Het is dan ook veel eenvoudiger en sneller om bestandskenmerken van bestanden te vergelijken, dan de inhoud van de bestanden zelf. Daarnaast is het veel eenvoudiger en sneller om te communiceren over bestandskenmerken van bestanden, dan over (de inhoud van) de bestanden zelf.

Bestandskenmerken helpen verder bij het identificeren van bestanden. Bij deze forensische toepassing wordt gebruik gemaakt van een database met bestandskenmerken van bekende, geclassificeerde bestanden. Een voorbeeld is het identificeren van bestanden met kinderpornografische afbeeldingen. Om een bestand te identificeren wordt eerst het bestandskenmerk van dat bestand berekend. Vervolgens gaat de onderzoeker na of dit bestandskenmerk voorkomt in de database. Is dit het geval, dan kijkt de onderzoeker hoe het bestandskenmerk is geclassificeerd. De database kan zowel bestandskenmerken bevatten van bestanden die eerder aangemerkt zijn als kinderpornografisch als van bestanden die bekende niet-relevante gegevens bevatten. Deze methode maakt het daarom mogelijk om zowel relevante bestanden te ontdekken als om niet-relevante bestanden vroegtijdig voor verder onderzoek uit te sluiten. Omdat het extreem onwaarschijnlijk is dat door deze methode verschillende bestanden hetzelfde bestandskenmerk zullen hebben, is de kans op een foutieve classificatie verwaarloosbaar klein (zo goed als 0). In paragraaf 7.3 wordt per door DBS gebruikt hashalgoritme rekenkundig aangegeven waarom de kans op een foutieve classificatie verwaarloosbaar klein is.

6. Bestandskenmerken in de praktijk

DBS gebruikt bestandskenmerken hoofdzakelijk om de integriteit van digitaal materiaal te controleren (bij aanvang onderzoek) of controleerbaar te maken (bij resultaten van het onderzoek).

Bij het controleren gaat het om de vaststelling of tijdens het transport van en naar DBS en tijdens het onderzoek zelf geen fouten zijn ontstaan in het aangeleverde materiaal of kopieën daarvan. Een praktijkvoorbeeld vormt het veiligstellen door bijvoorbeeld de politie van de gegevens die zijn opgeslagen op de harde schijf van een computer (van een verdachte). In dat geval wordt over de veiliggestelde gegevens - dat wil zeggen de volledige inhoud van de gekopieerde harde schijf - één of meerdere bestandskenmerken berekend. De kopie met de gegevens van de harde schijf wordt vervolgens samen met de bestandskenmerken aangeleverd bij de afdeling DBS. Een medewerker van DBS berekent opnieuw de bestandskenmerken en vergelijkt deze met de door politie aangeleverde bestandskenmerken. Bij gelijke bestandskenmerken wordt aangenomen dat tussentijds geen wijzigingen tijdens het transport zijn opgetreden. Wanneer blijkt dat een door DBS berekend bestandskenmerk anders is dan het aangeleverde bestandskenmerk, is er iets fout gegaan tijdens het transport. Dit zal worden teruggekoppeld aan de aanvrager.

Het is gewenst dat eenzelfde controle uitgevoerd wordt tijdens inbeslagname van digitaal materiaal. Tijdens het veiligstellen door bijvoorbeeld de politie van gegevens van een in beslaggenomen harde schijf worden één of meerdere bestandskenmerken over deze gegevens berekend. Wanneer alle gegevens veiliggesteld zijn, worden hierover opnieuw bestandskenmerken berekend. Beide series bestandskenmerken worden aansluitend met elkaar vergeleken. Ook hier geldt dat bij gelijke bestandskenmerken aangenomen wordt dat geen wijzigingen zijn opgetreden. Wanneer in een later stadium opnieuw bestandskenmerken over de gegevens aanwezig op dezelfde harde schijf worden berekend, kan ook nog gecontroleerd worden of de gegevens in de tussentijd wel of niet gewijzigd zijn. Bij het controleerbaar maken gaat het om de digitale onderzoeksresultaten die aan de opdrachtgever worden verstrekt op bijvoorbeeld een dvd-recordable of een transportschijf. Om in de toekomst de integriteit van de resultaten te kunnen controleren berekent DBS één of meerdere bestandskenmerken van elk bestand dat wordt verstrekt. In zaken waarbij het resultaat bestaat uit een klein aantal bestanden, rapporteert DBS soms de bestandskenmerken van alle bestanden. Als het resultaat bestaat uit een groot aantal bestanden, dan slaat DBS de bestandskenmerken van deze bestanden op in een nieuw bestand dat meestal *hashes.txt* (ook *bestandskenmerken.txt*) genoemd wordt. Vervolgens worden één of meerdere bestandskenmerken van het bestand hashes.txt zelf berekend. Deze laatste bestandskenmerken staan dan vermeld in een DBSrapport.

Figuur 1 geeft schematisch weer waar bestandskenmerken gebruikt (zouden moeten) worden in het proces van inbeslagname tot en met opleveren van resultaten.

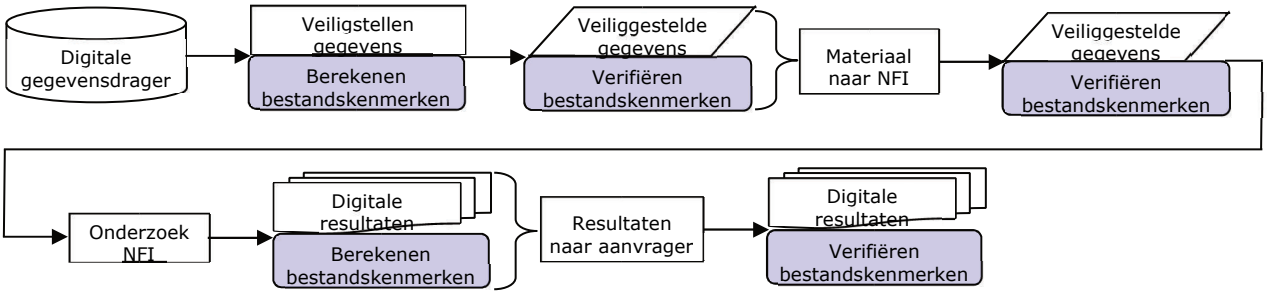
7. Door DBS gebruikte hashalgoritmen

In deze paragraaf wordt aangegeven wat de huidige status is van de gebruikte hashalgoritmen. Op basis van deze status wordt uitgelegd waarom welke algoritmen door DBS gebruikt worden. Aansluitend wordt per algoritme aangegeven wat de kans in de praktijk is op een ander bestand met hetzelfde bestandskenmerk.

7.1. Huidige status van hashalgoritmen

Voorbeelden van hashalgoritmen zijn MD4 en MD5 (`Message Digest') en de SHA (`Secure Hash Standard') series van gestandaardiseerde hashalgoritmen. De hashalgoritmen MD4 en SHA-0 zijn inmiddels gekraakt. In 1995 beval de Amerikaanse National Security Agency (NSA) aan om op het verbeterde hashalgoritme SHA-1 over te stappen omdat een zwakheid in SHA-0 was ontdekt. In 2004 hebben onderzoekers zwakheden in het veelgebruikte hashalgoritme MD5 ontdekt en gepubliceerd. Het werd daardoor mogelijk om twee bestanden met dezelfde MD5-bestandskenmerk te maken. Later is deze aanval op MD5 nog breder toepasbaar gemaakt. Hierdoor werd het relatief eenvoudig om twee verschillende, betekenisvolle bestanden (zoals leesbare documenten) te construeren met hetzelfde MD5-bestandskenmerk.
Sinds februari 2017 geldt hetzelfde voor SHA-1.

Het is van belang om te weten dat alleen de botsingbestendigheid van MD5 en SHA-1 gekraakt is. Echter, de in paragraaf 5 beschreven forensische toepassingen zijn onafhankelijk van deze eigenschap. Dit betekent dat tot nu toe MD5 en SHA-1 beide nog prima bruikbaar zijn voor integriteitscontrole van digitaal materiaal en identificatie en classificatie van bestanden. Voor sommige toepassingen is het een groot probleem wanneer een efficiënte methode beschikbaar komt om twee verschillende bestanden met hetzelfde bestandskenmerk te construeren. Zo'n *collision attack* kan in theorie ook een probleem vormen bij het classificeren van bestanden, namelijk wanneer een database met bestandskenmerken wordt gebruikt om oninteressante bestanden van onderzoek uit te sluiten. De maker van een dergelijk bestand kan dan misschien ook een ander, minder onschuldig bestand hebben gemaakt, dat toch hetzelfde bestandskenmerk heeft. In de praktijk valt dit probleem te ontwijken.



Figuur 1: schematische weergave van de controle van digitaal materiaal met behulp van bestandskenmerken

De oplossing is alleen bestanden van bekende en vertrouwde oorsprong als niet-relevant te classificeren in de database en geen bestanden van onbekende makers als niet-relevant te classificeren.

Een veel groter probleem voor forensische toepassingen is het, wanneer een efficiënte methode beschikbaar komt om een bestand te produceren dat hetzelfde bestandskenmerk heeft als een ander, vooraf gegeven, bestand. Een dergelijke methode voor zo'n *second preimage attack* is nog niet bekend voor de gangbare algoritmen MD5 en SHA-1, maar de vakliteratuur beschrijft wel voortgang in het zoeken hiernaar. Een preventieve overstap naar één van de veiliger geachte SHA-2 hashalgoritmen is daarom aan te bevelen voor forensische toepassingen. Afgezien van enkele uitzonderingen is DBS medio 2010 overgestapt op het SHA-2 algoritme met een lengte van 256 bits (ook wel SHA-256 genoemd).

7.2. Huidige keuze hashalgoritmen bij het DBS

De wereldwijd meest gebruikte hashalgoritmen om de integriteit van digitale gegevens te controleren en voor bestandsidentificatie zijn MD5 met een lengte van 128 bits en SHA-1 met een lengte van 160 bits. Zowel binnen als buiten de forensische wereld vinden ze veel toepassing. De veiligheidsgaranties die ze bieden zijn, ondanks voortdurende 'aanvallen' op deze algoritmes, nog steeds extreem hoog (zie voor numerieke definities paragraaf 7.3) voor integriteitscontrole en bestandsidentificatie. Zoals eerder vermeld is DBS in 2010 uit voorzorg overgestapt op het SHA-256 algoritme. DBS moet echter in de praktijk nog steeds vaak bestandskenmerken verifiëren die ketenpartners aanleveren. Hierbij worden (ook vanwege de door de ketenpartners gebruikte programmatuur) nog steeds MD5 en SHA-1 aangeleverd.

In de periode 2010-2012 liep via het Amerikaanse *National Institute for Standards and Technology* (NIST) een openbare wedstrijd om een nog beter nietomkeerbaar hashalgoritme SHA-3 te kiezen. Op 2 oktober 2012 is het winnende algoritme gekozen, genaamd *Keccak*. Ten tijde van de laatste aanpassing van deze vakbijlage is het gebruik van SHA-3-Keccak in de forensische wereld nog geen gemeengoed. Aangezien SHA-256 voorlopig nog veilig genoeg is, is DBS dan ook nog niet overgegaan op SHA-3-Keccak.

7.3. Kans in de praktijk op ander bestand met hetzelfde bestandskenmerk.

De kans dat een willekeurig (ander) bestand hetzelfde bestandskenmerk heeft als een bepaald *gegeven* bestand wordt hier uitgelegd aan de hand van het volgende voorbeeld. Neem een database met bestandskenmerken van bestanden waarvan de inhoud geclassificeerd is als kinderpornografisch. Stel nu dat in een onderzoek bestand met de naam `B` wordt aangetroffen, waarbij de bestandskenmerken van `B` voorkomen in deze database. Wat is nu de kans dat `B` toevallig toch een andere inhoud heeft dan het kinderpornografische bestand? Dit wordt hier uitgerekend per gebruikt bestandskenmerk.

- MD5
  Een MD5-bestandskenmerk bestaat uit 128 bits. Ervan uitgaande dat elk MD5-bestandskenmerk met dezelfde kans voor kan komen zijn er in totaal
  2^{128} (≈ 3,40 x 10^{38}) aan mogelijke MD5-bestandskenmerken. Dit betekent dat de kans dat een willekeurig ander bestand met andere inhoud
  toch hetzelfde MD5- bestandskenmerk heeft gelijk is aan $\frac{1}{2^{128}} \approx 2{,}9\ x\ 10^{-39}$.
- SHA-1
  Een SHA-1- bestandskenmerk bestaat uit 160 bits. Een vergelijkbare berekening als bij de MD5- bestandskenmerk geeft dat de kans dat een willekeurig ander bestand met andere inhoud toch
  dezelfde SHA-1- bestandskenmerk heeft gelijk is aan $\frac{1}{2^{160}} \approx 6{,}8\ x\ 10^{-49}$.
- SHA-256
  Een SHA-256 bestandskenmerk uit 256 bits. Een vergelijkbare berekening als bij de MD5- en SHA-1- bestandskenmerken geeft dat de kans dat een willekeurig ander bestand met andere inhoud toch
  dezelfde SHA-256- bestandskenmerk heeft gelijk is aan $\frac{1}{2^{256}} \approx 8{,}6\ x\ 10^{-78}$.

Met de huidige stand van de techniek kan op basis van deze berekeningen voor elk van de door DBS gebruikte bestandskenmerken gezegd worden dat de kans dat een bestand met andere inhoud dan een bekend gegeven bestand toch hetzelfde bestandskenmerk heeft extreem klein (zo goed als nul) is.

Ter vergelijking: de theoretische *random match probability* van het *Next-Generation Multiplex* (NGM) systeem in de Nederlandse populatie (gebaseerd op 15 loci) is op minimaal $1{,}0\ x\ 10^{-25}$. Uit het voorgaande blijkt dat de kans dat twee inhoudelijk verschillende bestanden bij toeval hetzelfde bestandskenmerk MD5, SHA-1 en/of SHA-256 hebben gelijk is aan of kleiner is dan $2{,}9\ x\ 10^{-39}$. Deze laatste kans is dus extreem veel kleiner dan dat twee verschillende mensen bij toeval een niet te onderscheiden DNA-profiel hebben.

8. Verklarende woordenlijst

- Algoritme
  Een eindige reeks instructies om vanuit een gegeven begintoestand het daarbij behorende duidelijk beschreven eindresultaat te bereiken (naar de naam van de Arabische wiskundige AlChwarizmi). Een algoritme kan behulp van een computertaal worden geïmplementeerd tot een computerprogramma, en zo door een computer (automatisch) worden uitgevoerd. Een algoritme kan vergeleken worden met een kookrecept: er moeten bepaalde stappen worden gevolgd om tot een (gewenst) eindresultaat te komen. Een extreem simpel voorbeeld van een (rekenkundig) algoritme om uit te rekenen hoeveel keer je een getal bij zichzelf moet optellen om op of boven de 100 te komen is het volgende:

```
Stap 1: vraag een getal
Stap 2: aantal keer = 0
Stap 3: tussenresultaat = getal Stap 4:
als het tussenresultaat 100 of hoger is,
ga naar stap 8 Stap 5: verhoog het
tussenresultaat met het getal
Stap 6: verhoog het aantal keer met 1
Stap 7: ga terug naar stap 4
Stap 8: geef het aantal keer
```

  Wanneer bij de eerste stap ‘7’ wordt gegeven, zal het antwoord (‘aantal keer’) gelijk zijn aan ‘14’.

- Bits en Bytes
  Een bit kan de waarde ‘0’ of ‘1’ hebben. Een byte bestaat uit 8 bits. Omdat elke bit twee waarden (‘0’ of ‘1’) kan hebben, heeft een byte 2^8 = 256 mogelijke waarden. Bytes worden vaak in hexadecimale notatie geschreven (voorafgegaan door 0x om onderscheid te kunnen maken met gewone decimale getallen). De waarde van een byte varieert hexadecimaal dan ook tussen 0x00 en 0xFF.

9. Literatuurlijst

- Rivest, R.L. *The MD4 Message Digest Algorithm*, Crypto '90 Proceedings, 1991.
- Rivest, R.L. *The MD5 Message-Digest Algorithm*, Request For Comments (RFC) 1321, Internet Activities Board, Internet Privacy Task Force, 1992.
- FIPS 180-1, *Secure hash standard*, National Institute for Standards and Technology (NIST), US Department of Commerce, Washington D.C., April 1995. Springer-Verlag, 1996.
- Menezes, A.J., Oorschot, P.C. van, Vanstone, S.A. *Handbook of Applied Cryptography*, CRC Press, 1997.
- Chabaud, F., Joux, A. *Differential Collisions in SHA-0*, Advances in Cryptology, Springer Verlag, LNCS 1462, 1998.
- Wang, X.Y., Guo, F.D., Lai, X.J., Yu, H.B. *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, Rump Session of Crypto'04, Eprint, 2004
- Weger, B.M.M. de, *Hash-functies onder vuur, de situatie van MD5 en SHA-1*, PvIB (Platform voor Informatie Beveiliging), nummer 2, pagina's 25-30, 2005
- Wang, X.Y., Lai, X.J., Feng, D., Chen, H., Yu, X.Y. *Cryptanalysis of the Hash Functions MD4 and RIPEMD*, Advances in Cryptology, Springer Verlag, LNCS 3494, 2005.
- Hoffman, P. Schneier, B. *Attacks on Cryptographic Hashes in Internet Protocols*, Request For Comments (RFC) 4270, Internet Engineering Task Force, 2005.
- Joux, A. *Multicollisions in iterated hash functions. Application to cascaded constructions*, DCSSI Crypto Lab, France, 2005.
- Biham, E., Chen, R., Joux, A., Carribault, P., Lemuet, C., Jalby, W. *Collisions of SHA-0 and Reduced SHA-1*, IACR paper volume 3494, pagina' 36-57, 2005.
- Wang, X.Y., Yu, H.B. *How to Break MD5 and Other Hash Functions*, Advances in Cryptology EuroCrypt 2005, Springer Verlag, pagina's 19-35, 2005.
- Aumasson, J.P., Meier W., Mende, F. *Preimage Attacks on 3-Pass HAVAL and Step-Reduced MD5*, Cryptology ePrint Archive, Report 2008/183, 2008.
- Cannière, C., Rechberger, C. *Preimages for Reduced SHA-0 and SHA-1*. Advances in Cryptology, Springer Verlag, LNCS 5157, 2008.
- Meulenbroek, A.J. *De essenties van forensisch biologisch onderzoek – Humane biologische sporen en DNA*, uitgeverij Paris, hoofdstuk 7 (pagina's 157 – 176), 5e druk, 2009.
- Informatieblad *DNA-verwantschapsonderzoek*, versie 1, NFI.
- Bertoni, G., Daemen, J., Peeters, M., Assche, G. van, *The Keccak reference*, version 3.0, 2011 (www.keccak.noekeon.org).
- NIST, *SHA-3 Competition (2007-2012)*, http://csrc.nist.gov/groups/ST/hash/sha3/index.html.
- Clark, R.A., Morell, M.L., Stone, G.R., Sunstein, C.R., Swirre, P. *The NSA Report*, Princeton University Press, 2013.
- Stevens, M., Bursztein, E., Karpman, P., Albertini, A., Markov, Y. *The first collision for full SHA-1*, preprint 2017 (https://shattered.io/).
- Leurent, G., Peyrin, T. *From Collisions to ChosenPrefix Collisions Application to Full SHA-1*, Advances in Cryptology EUROCRYPT 2019, april 2019 (https://eprint.iacr.org/2019/459.pdf).
- Leurent, G., Peyrin, T. *SHA-1 is a Shambles*, USENIX Security '20, augustus 2020, (https://eprint.iacr.org/2020/014.pdf).

Voor algemene vragen kunt u contact opnemen met de Frontdesk, telefoon (070) 888 68 88. Voor inhoudelijke vragen kunt u contact opnemen met de Divisie Digitale en Biometrische Sporen, telefoon (070) 888 6400.

Nederlands Forensisch Instituut
Ministerie van Veiligheid en Justitie
Postbus 24044 | 2490 AA Den Haag

Telefoon (070) 888 66 66
www.forensischinstituut.nl