TEMA: SIMPLIFIED SHA-1

LËNDA: SIGURIA E TË DHËNAVE

UNIVERSITETI I PRISHTINËS

PUNOI:
BLEON BALAJ
ELJON KASTRATI

2023

PËRMBAJTJA

ÇKA ËSHTË SHA-1

SHA-1 (Secure Hash Algorithm 1) është një funksion hash kriptografik që i përket familjes SHA (Secure Hash Algorithm). Ai u zhvillua nga Agjencia Kombëtare e Sigurisë (NSA) dhe u botua nga Instituti Kombëtar i Standardeve dhe Teknologjisë (NIST) në 1995. SHA-1 përdoret gjerësisht për qëllime të ndryshme, duke përfshirë verifikimin e integritetit të të dhënave, nënshkrimet dixhitale dhe ruajtjen e fjalëkalimeve.

Algoritmi SHA-1 përpunon mesazhet hyrëse në blloqe prej 512 bitësh (64 bajt). Nëse mesazhi është më i gjatë, ai ndahet në blloqe dhe përpunohet në mënyrë sekuenciale. Algoritmi kryen hapat e mëposhtëm:

- Mbushja: Mesazhi i hyrjes është i mbushur për të siguruar që gjatësia e tij të jetë shumëfish i 512 biteve. Mbushja përfshin shtimin e një 1-biti të ndjekur nga zero, dhe më pas shtimin e gjatësisë së mesazhit si një paraqitje 64-bitësh.
- Inicializimi: Algoritmi inicializon një vektor të gjendjes 160-bit me vlera të paracaktuara.
- Orari i mesazheve: Blloku i hyrjes ndahet në fjalë 32-bit, duke rezultuar në një grup të programeve të mesazheve prej 80 fjalësh. Fjalët shtesë nxirren duke zbatuar veprime bitwise në fjalët e përpunuara më parë.
- Llogaritja e hash: Orari i mesazheve përdoret në një seri raundesh për të përditësuar vektorin e gjendjes. Çdo raund përbëhet nga operacione në bit dhe funksione logjike të aplikuara në gjendjen aktuale dhe fjalën përkatëse nga programi i mesazheve.
- Finalizimi: Pas përpunimit të të gjitha blloqeve të mesazheve, vektori i gjendjes që rezulton përfaqëson vlerën hash SHA-1.

HASH FUNKSIONET

Një funksion hash është një funksion matematik që merr një hyrje (mesazh) dhe prodhon një dalje me madhësi fikse (vlerë hash), e përfaqësuar zakonisht si një sekuencë shifrash ose karakteresh. Funksionet hash kanë disa veti të rëndësishme:

- Deterministik: I njëjti input do të prodhojë gjithmonë të njëjtin dalje.
- Llogaritja e Shpejtë: Hashimi duhet të jetë efikas nga ana llogaritëse.
- Rezistenca e paraimazhit: Duke pasur parasysh një vlerë hash, duhet të jetë llogaritëse e pamundur të përcaktohet hyrja origjinale.

SIGURIA

Ndërsa SHA-1 përdorej gjerësisht në të kaluarën, tani konsiderohet i pasigurt për shumë aplikacione kriptografike për shkak të përparimeve në fuqinë llogaritëse dhe kriptanalizën. Në vitin 2005, studiuesit demonstruan sulme teorike të përplasjes në SHA-1, dhe në vitin 2017, një sulm praktik përplasjeje u ekzekutua me sukses. Rrjedhimisht, përdorimi i SHA-1 dekurajohet fuqimisht në sistemet dhe aplikacionet e reja që kërkojnë siguri të fortë.

Për të siguruar një hash të sigurt dhe integritet të të dhënave, rekomandohet përdorimi i funksioneve hash më të sigurta, si SHA-256 ose SHA-3, të cilat ofrojnë garanci më të forta sigurie. Këto algoritme janë pjesë e familjeve SHA-2 dhe SHA-3, respektivisht, dhe ofrojnë madhësi më të mëdha hash dhe rezistencë të përmirësuar kundër sulmeve.

Kur migroni nga SHA-1 në një funksion hash më të sigurt, është e rëndësishme të vlerësoni me kujdes kërkesat specifike të sistemit dhe të zgjidhni një algoritëm të përshtatshëm që i plotëson këto kërkesa.

PËRPARËSITË

- Efikasiteti: SHA-1 është relativisht i shpejtë dhe efikas për sa i përket llogaritjes, duke e bërë atë të përshtatshëm për aplikacione ku performanca është një përparësi. Mund të përpunojë mesazhe në blloqe prej 512 bitësh, duke siguruar një ekuilibër midis shpejtësisë dhe sigurisë.
- Përputhshmëri e gjerë: SHA-1 është implementuar dhe mbështetur gjerësisht nga gjuhë të ndryshme programimi, biblioteka dhe sisteme. Përdorimi i gjerë i tij në të kaluarën ka çuar në integrimin e tij në shumë sisteme dhe protokolle ekzistuese, duke e bërë atë të pajtueshëm me një gamë të gjerë aplikacionesh dhe platformash.

MANGESITË

- Dobësitë ndaj sulmeve të përplasjes: Një nga të metat kryesore të SHA-1 është cenueshmëria e tij ndaj sulmeve të përplasjes. Me kalimin e viteve, përparimet në fuqinë llogaritëse dhe kriptanalizën e kanë bërë të mundur gjetjen e dy hyrjeve të ndryshme që prodhojnë të njëjtën vlerë hash SHA-1. Kjo rrezikon integritetin e algoritmit, pasi ai nuk ofron më nivelin e dëshiruar të sigurisë.
- Siguria e vjetëruar: Për shkak të dobësive të demonstruara dhe sulmeve praktike të përplasjes në SHA-1, ajo tani konsiderohet e pasigurt për qëllime kriptografike. Ekspertët dhe organizatat e sigurisë dekurajojnë fuqimisht përdorimin e tij në sisteme të reja dhe rekomandojnë kalimin në funksione hash më të forta, si SHA-256 ose SHA-3, të cilat ofrojnë garanci të zgjeruara sigurie dhe rezistencë ndaj sulmeve të avancuara.

IMPLEMENTIMI I ALGORITMIT NË GJUHËN JAVA

```java
public class SHA1 {
    public static void main(String[] args) {
        String input = …; // Mesazhi
        String sha1Hash = sha1(input);
        System.out.println("SHA-1 Hash: " + sha1Hash);
    }

    public static String sha1(String input) {
        int[] words = prepareMessage(input);
        int[] state = initialState();

        int[] messageSchedule = new int[80];
        for (int i = 0; i < words.length; i += 16) {
            int[] w = new int[80];
            System.arraycopy(words, i, w, 0, 16);

            for (int t = 16; t < 80; t++) {
                w[t] = leftRotate(w[t - 3] ^ w[t - 8] ^ w[t - 14] ^ w[t - 16], 1);
            }

            int[] temp = new int[5];
            System.arraycopy(state, 0, temp, 0, 5);

            for (int t = 0; t < 80; t++) {
                int f, k;
                if (t < 20) {
                    f = (temp[1] & temp[2]) | (~temp[1] & temp[3]);
                    k = 0x5A827999;
                } else if (t < 40) {
                    f = temp[1] ^ temp[2] ^ temp[3];
                    k = 0x6ED9EBA1;
                } else if (t < 60) {
                    f = (temp[1] & temp[2]) | (temp[1] & temp[3]) | (temp[2] & temp[3]);
                    k = 0x8F1BBCDC;
                } else {
                    f = temp[1] ^ temp[2] ^ temp[3];
                    k = 0xCA62C1D6;
                }

                int tempVal = leftRotate(temp[0], 5) + f + temp[4] + k + w[t];
                temp[4] = temp[3];
                temp[3] = temp[2];
                temp[2] = leftRotate(temp[1], 30);
                temp[1] = temp[0];
                temp[0] = tempVal;
            }

            for (i = 0; i < 5; i++) {
                state[i] += temp[i];
            }
        }

        StringBuilder hexString = new StringBuilder();
        for (int val : state) {
            hexString.append(String.format("%08x", val));
        }

        return hexString.toString();
    }

    private static int[] prepareMessage(String input) {
        byte[] message = input.getBytes();
        int messageLength = message.length;
        int paddedLength = ((messageLength + 8) / 64 + 1) * 64;

```

```
66          byte[] paddedMessage = new byte[paddedLength];
67          System.arraycopy(message, 0, paddedMessage, 0, messageLength);
68          paddedMessage[messageLength] = (byte) 0x80;
69
70          long messageBits = (long) messageLength * 8;
71          for (int i = 0; i < 8; i++) {
72              paddedMessage[paddedLength - 8 + i] = (byte) (messageBits >>> (56 - i * 8));
73          }
74
75          int[] words = new int[paddedLength / 4];
76          for (int i = 0; i < words.length; i++) {
77              int word = 0;
78              for (int j = 0; j < 4; j++) {
79                  word |= (paddedMessage[i * 4 + j] & 0xFF) << (24 - j * 8);
80              }
81              words[i] = word;
82          }
83
84          return words;
85      }
86
87      private static int[] initialState() {
88          int[] state = new int[5];
89          state[0] = 0x67452301;
90          state[1] = 0xEFCDAB89;
91          state[2] = 0x98BADCFE;
92          state[3] = 0x10325476;
93          state[4] = 0xC3D2E1F0;
94          return state;
95      }
96
97      private static int leftRotate(int value, int shift) {
98          return (value << shift) | (value >>> (32 - shift));
99      }
100 }
```

KONKLUZIONI

SHA-1 është një funksion hash kriptografik i përdorur gjerësisht që është përdorur gjerësisht në aplikacione të ndryshme. Megjithatë, për shkak të dobësive të tij ndaj sulmeve të përplasjes, ai nuk konsiderohet më i sigurt për aplikacionet që kërkojnë siguri të fortë. Rekomandohet fuqimisht kalimi në funksione hash më të sigurta, si SHA-256 ose SHA-3, për të siguruar integritetin e të dhënave dhe forcën kriptografike.