**5-TOPSHIRIQ**

***Iqbolshoh Ilhomjonov***

**2. Xesh funksiyalarning qanday turlari mavjud?**

Boshqa xash turlari quyidagilardir:

1. SHA-1 (Secure Hash Algorithm 1)

2. SHA-2 (Secure Hash Algorithm 2) - misol uchun SHA-256, SHA-384, va SHA-512

3. SHA-3 (Secure Hash Algorithm 3)

4. BLAKE (Blake2b and Blake2s)

5. Whirlpool

Barcha xash turlari shifrlash va xavfsizlik sohasida foydalaniladi. Xashlar, ma'lum bir ma'lumotni o'z ichiga olib, o'zgarisiz yoki qaytarilishi murakkab bo'lmagan unikal xujjatlarga aylantirish uchun ishlatiladi. Misol uchun, parolni saqlashda foydalanish uchun matnni shifrlashda va yashirishda foydalaniladi.

**4. Kriptografik xesh funksiya strukturasi va unga misollar keltiring ?**

Kriptografik xesh funkisiyalar hash (ya'ni struktur) funksiyalariga kiradi. Hash funksiyalari kelgusi xabar yoki faylni xatoliklar va yozma amalga oshirilgan qiymatlarning ko’rsatilgan katta kichikligi bo’yicha bitta turli ko’rsatkichga (hash, checksum yoki digest) o’zgartirishga olib keladi. Xesh funksiyalari tarqatiladigan maqsadlar orasida dasturlarni tasdiqlash, tarmoq xavfsizligini ta’minlash, foydalanuvchilarni identifikatsiya qilish, sifatli ma’lumotlarni ishonchli ravishda saqlash va h.k. kiritiladi.

Kriptografik xesh funkisiyalariga qo’shimcha talablar qo’yiladi, masalan, ma'lumotlarning avvaldan aniqlangan va unga tesir ko'rsatmasligi, bir tomondan boshqa ma'lumotlar avvaldan aniqlangan holda aniqlanmasligi, bir necha kirish ko’rsatmalariga mos kelishligi kabi.

Misol uchun, quyidagi kriptografik xesh funkisyalariga misollar keltiriladi:

1. MD5 (Message-Digest Algorithm 5): 128-bit uzunligidagi xesh qiymatini hosil qiladi. Eski va hatto boshqa kishilar tomonidan ham ishlatilgan.

2. SHA-1 (Secure Hash Algorithm 1): 160-bit uzunligidagi xesh qiymatini hosil qiladi. Bu funksiya ham ko’p kishi tomonidan ishlatiladi.

3. SHA-256, SHA-384, va SHA-512: Farqli uzunlikdagi xesh qiymatlari hosil qilish uchun ishlatiladi. Bu funksiyalar hozirgi kunda ko’p ishlatiladi va xavfsizlik so’rovlarida juda kuchli xavfsizlikni ta’minlaydi.

**6. Kiruvchi ma’lumotlarni polinimal xesh funsiya yordamida xeshlang.**

Kiruvchi ma'lumotlarni polinomial xesh funksiyalari yordamida xeshlash uchun quyidagi algoritm amalga oshiriladi:

1. Kiruvchi ma'lumotlar (matn, fayl yoki boshqa turdagi ma'lumotlar) olinadi.

2. Kiruvchi ma'lumotlar polinomial xesh funksiyasi orqali bir xesh qiymatiga aylantiriladi.

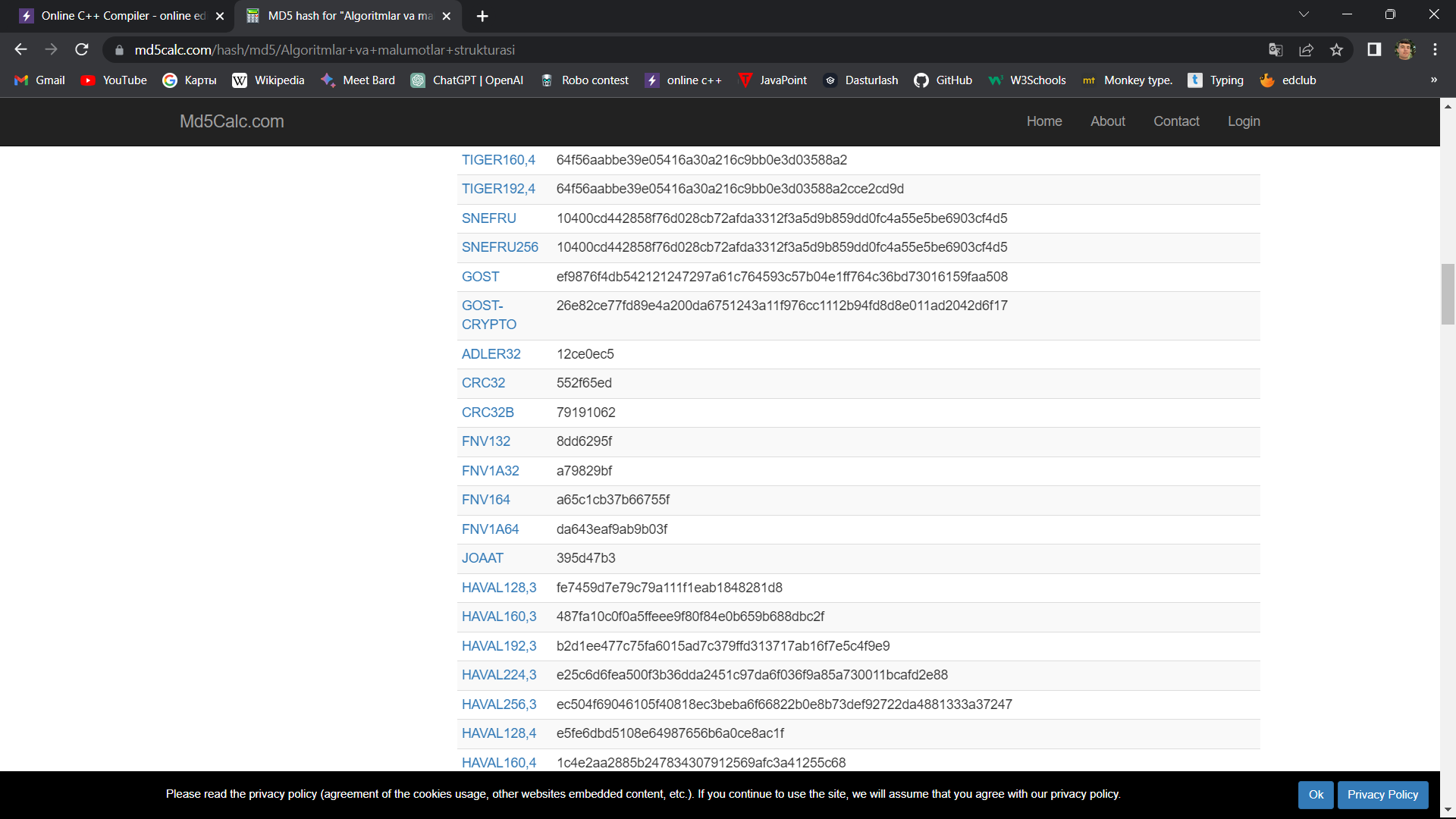
3. Aylantirilgan xesh qiymati, ma'lumotlar bilan birga saqlanadi yoki uni yuborish uchun foydalaniladi.

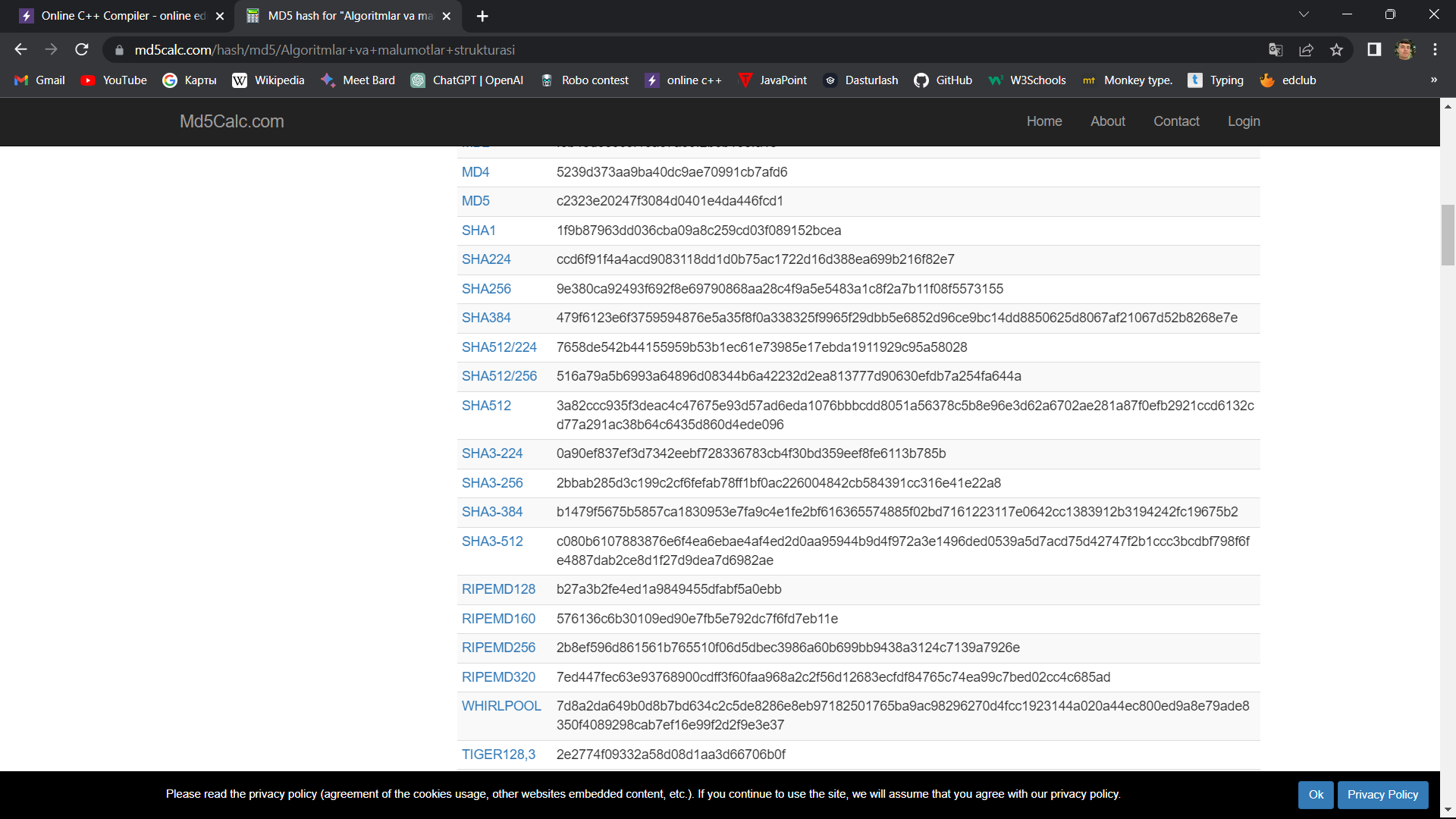
4. Qayta ishlovchi ma'lumotlar kelgusi kiritilganida, yangi xesh qiymati qayta hosil qilinadi va uni oldingi qiymat bilan solishtiriladi.

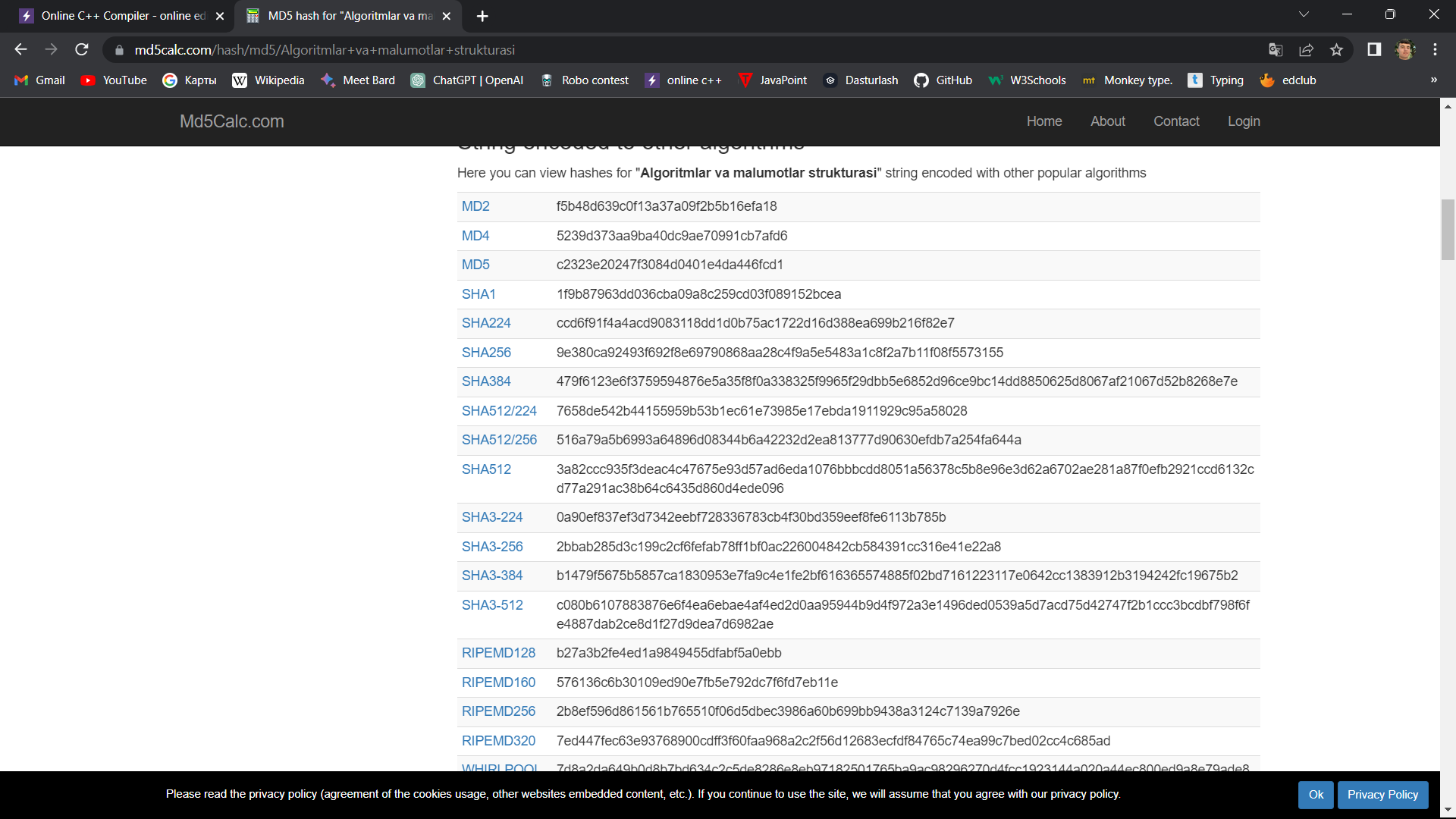
5. Agar yangi xesh qiymati oldingi xesh qiymati bilan mos kelmasa, ma'lumotlarda o'zgarish bo'lishi va uning o'zgartirilgan xesh qiymatida ham o'zgarish bo'lishi mumkin.

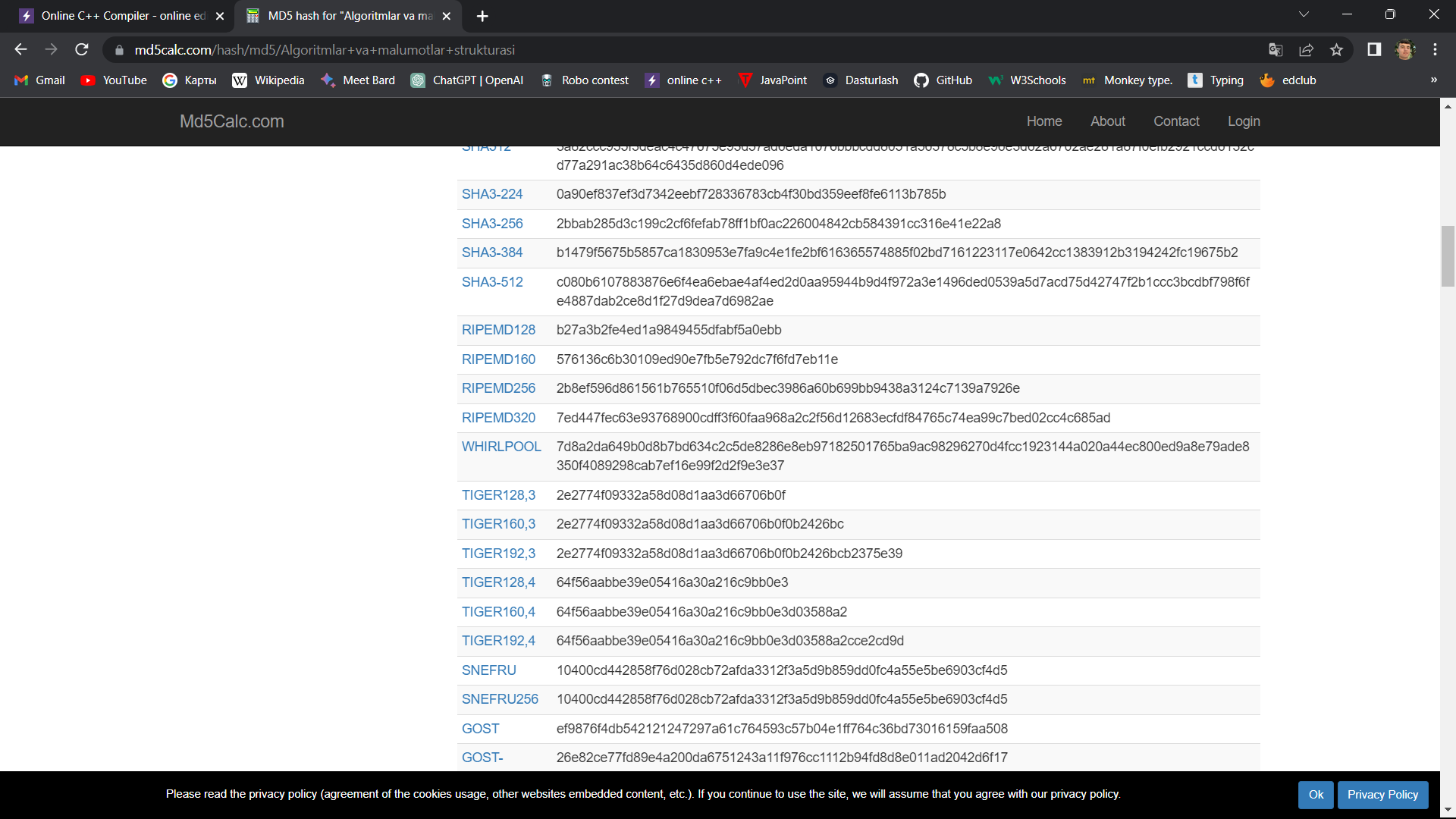
Kriptografik xesh funksiyalari, masalan, MD5 va SHA-1 polinomial xesh funksiyalari mavjud. Bu funksiyalar shunchaki ma'lumotlarning xavfsizligi uchun katta ahamiyatga ega.

**8. “ Algoritm va ma’lumotlar strukturasi” matnini Adler-32, SHA-1, MD-4, MD-5, Tiger xesh funksiyalari yordamida xeshlang.**









**10. Quyidagi dasturni tahlil qiling.**

#include <iostream>

#include <string.h>

using namespace std;

int main()

{

    char a[100];

    cout<<"So'zni kiriting: = ";

    cin.getline(a,100);

    cout<<"Shifrlangan kalit ";

    for(int i=0,j=strlen(a)+67;

    j<=strlen(a),i<=strlen(a);i++,j=j+3)

    cout<<char (j)  <<j <<(int)a[i]+55+122\*2<<char(j);

    cout<<endl;

    main();

return 0;

}

Bu dastur foydalanuvchidan bir so'z kiritishni so'raydi va shu so'zni shifrlaydi. Shifrlash usuli esa har bir harfning unikod qiymatini o'z ichiga oladi va shu qiymatlardan foydalanib shifrlangan kalitni hosil qiladi. Shifrlangan kalitda harfning unikod qiymati, uni 55 ga qo'shish, 122 ga ko'paytirish va 2 ga ko'paytirishdan hosil qilingan qiymatlar ko'rinishida chiqariladi. Shifrlangan kalitda harfning unikod qiymati, uning qaysi tartibda kiritilganligiga bog'liq bo'lib, shunchaki bir nechta raqamlar, simvollar va xarflar orqali ajratilgan.

**12. Polinimal xeshlash algoritm struktuarsini izohlang.**

#include <iostream>

#include <string.h>

using namespace std;

long long Heshlash(char s[])

{

    long long h = 0;

    int base = 37;

    for(int i=0; i<=strlen(s); i++)

    {

        h = h\* base + s[i] - 61 +1;

    }

    return h;

}

int main()

{

    char s[100];

    for(int i=1; i<10; i++)

    {

    cin.getline(s,100);

    cout << s << " " << Heshlash(s);

    cout << endl;

 }

}

Bu dasturda "polinom heshlash" algoritmi implementatsiyalangan. Bu algoritm ma'lumotlarning qisqa ko'rsatkichlarini yaratishda qo'llaniladi. Algoritm ma'lumotlarni "char" turida qabul qiladi va har bir "char" belgisi uchun bir "long long" ko'rsatkich qiymatini qaytaradi.

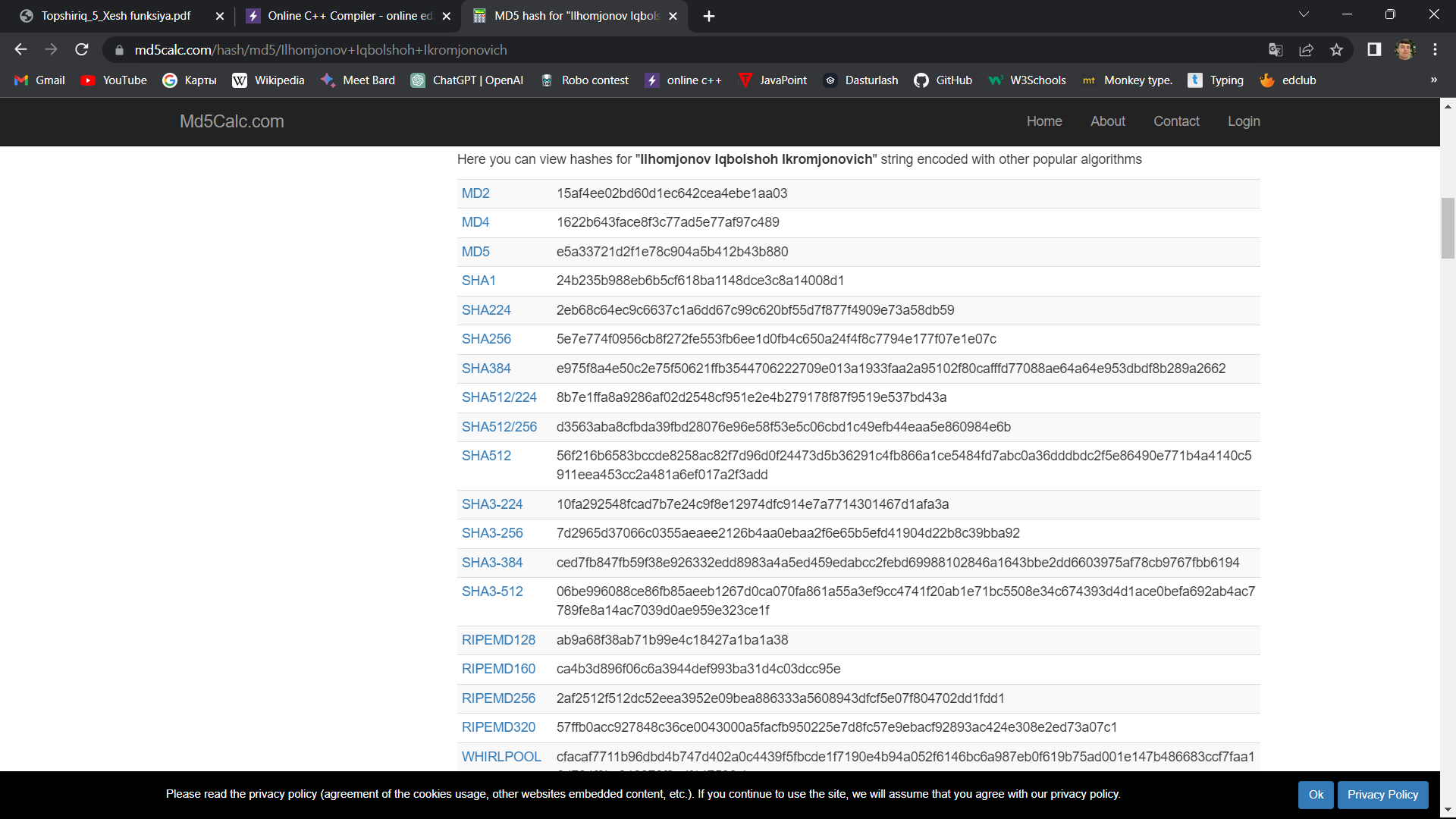
Algoritmning amaliyoti quyidagicha bo'ladi: Har bir belgi (harf, raqam yoki belgi) uchun, uning qiymatini belgilangan o'sish darajasi (polinom o'sishi) bilan ko'paytiriladi. Polinom darajasi ma'lumotlar uchun muhimdir, chunki uchta belgi uchun ko'paytirilgan qiymat bir-biridan farqli bo'lishi mumkin. Polinomdarajasi har xil bo'lishi mumkin, ammo yuqori darajalarni yaratish mumkinligi bor.

Polinom heshlash algoritmi murakkablik darajasi 1 dan 2 ga teng polinomlar uchun idealdir. Bu odatda yuqori darajali polinomlar uchun ishlatilmaydi.

Yuqoridagi dasturda "37" soni polinomning darajasi sifatida ishlatilgan. Polinom heshlash algoritmini ko'proq tushunish uchun, belgilangan darajaga mos keluvchi boshqa sonlar ham o'zgartirilishi mumkin.

**14. Online xesh kalkulyatori yordamida kriptografik turli xil hesh funksiyalardan foydalanib, Ism, familyangiz, va sharfingizni xeshlang.**

Ilhomjonov Iqbolshoh Ikromjonovich



**16. Polinimal xesh funksiya dasturini tuzing.**

#include <iostream>

#include <string>

using namespace std;

int polinomial\_hash(string s) {

    int p = 31, m = 1e9 + 9; // modul qilinadigan son

    int hesh\_qiymat = 0, p\_pow = 1;

    for (char c : s) {

        hesh\_qiymat = (hesh\_qiymat + (c - 'a' + 1) \* p\_pow) % m;

        p\_pow = (p\_pow \* p) % m;

    }

    return hesh\_qiymat;

}

int main() {

    string s;

    cout << "Matn kiriting: ";

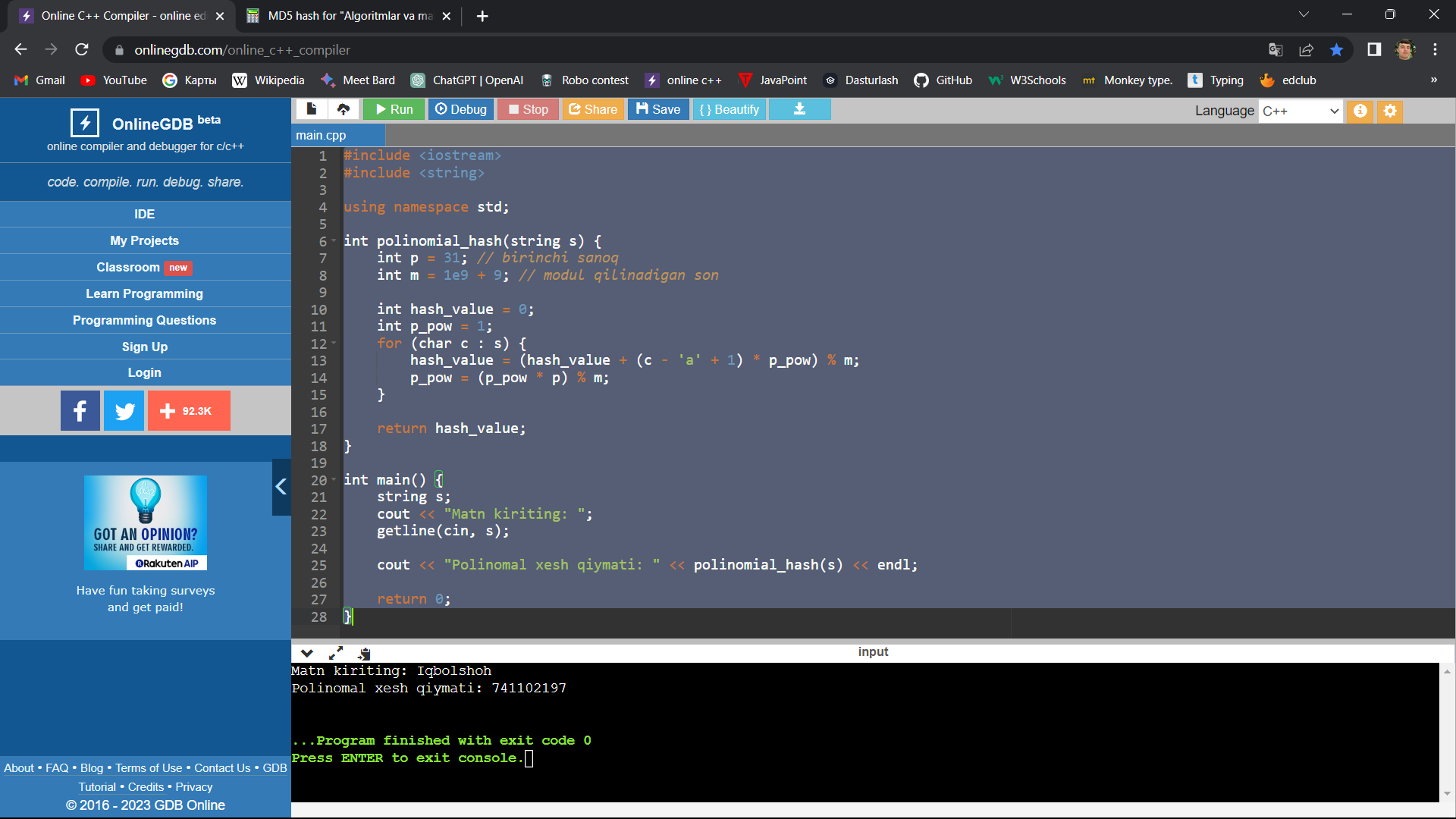
    getline(cin, s);

    cout << "Polinomal xesh qiymati: " << polinomial\_hash(s) << endl;

    return 0;

}

//NATIJA:



**20. Xesh funksiyalarning murakkabligini tahlil qiling. Yaxshi xesh funksiya qanday talablarga javob berishi lozim?**

Xesh funksiyalari, verilgan bir veri blokining saboqliligini aniqlash uchun ishlatiladi. Bu funksiyalar, hujjatlarning integral va to'liqligini tekshirish uchun, bu hujjatlarning boshqalar tomonidan almashtirilmaganligi va uning manzili o'zgartirilmaganligi kabi talablarga javob berishi lozim.

Xesh funksiyalari murakkabligi, ularga kiritiladigan ma'lumotlar kengligi va turiga bog'liq. Masalan, odatda, qisqa xatoliklarga olib kelish va murakkablik orasida tanlov kerak. Xesh funksiyalari murakkabligi ko'proq bo'lgan joylarda, ularni sifatli holatda ishlatish uchun yuqori tezlik va xavfsizlik ta'minlash uchun optimallashtirish talab etiladi.

Yaxshi xesh funksiyalari quyidagi talablarga javob berishi kerak:

1. Tekshiruvchi ma'lumotlar uchun yuqori darajada to'liqlik: Xesh funksiyasi, kiritilgan ma'lumotlar uchun xavfsizlikni ta'minlash uchun muhim bo'lib, tekshiruvchi ma'lumotlar (hash) uchun yuqori darajada to'liqlik ta'minlashi kerak.

2. Odatda, har bir kiritilgan ma'lumotning avvaldan aniqlangan holda aniqlanmasligi: Xesh funksiyasi, bir nechta kiritilgan ma'lumotlarni ko'rsatmasligi va bu ma'lumotlarning manzillarini o'zgartirmasligi kerak.

3. Xavfsizlik: Xesh funksiyasi, salbiy maqsadlar uchun yaratilgan kriptografik algoritmlardan biri bo'lib, yuqori darajada xavfsizlik ta'minlashi kerak.