ISSN:2181-0427 ISSN:2181-1458

ЎЗБЕКИСТОН РЕСПУБЛИКАСИ ОЛИЙ ВА ЎРТА МАХСУС ТАЪЛИМ ВАЗИРЛИГИ

НАМАНГАН ДАВЛАТ УНИВЕРСИТЕТИ ИЛМИЙ АХБОРОТНОМАСИ

НАУЧНЫЙ ВЕСТНИК НАМАНГАНСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА

2021 йил махсус сон

PYTHON TILIDA XABAR DAYJESTLARI BILAN ISHLASH

Otaxanov Nurillo Abdumalikovich
NamDU professori.

Annotatsiya. *Ushbu maqolada Python dasturlash tilida axborotlarni xesh-funksiyalar yordamida qayta ishlash jarayoni ochib berilgan. Unda xesh-funksiyalar bilan ishlashda foydalanish mumkin bo'lgan shifrlash algoritmlari batafsil bayon etilgan. Shuningdek, maqolada keltirilgan nazariy ma'lumotlarni amaliyotga tatbiq etish namunalari ham ko'rsatilgan.*

Kalit so'zlar: *Python, ma'lumot, bit, kod, algoritm, funksiya, modul, hash, hashlib.*

РАБОТА С ДАЙЖЕСТАМИ СООБЩЕНИЙ НА ЯЗЫКЕ PYTHON

Отаханов Нурилло Абдумаликович,
профессор НамГУ.

Аннотация. *В данной статье рассказывается о процессе обработки различных данных с помощью хеш-функций. В ней подробно описаны алгоритмы шифрования, используемые в работе с хеш-функциями. А также, показаны образцы применения на практике приведённых теоретических материалов.*

Ключевые слова: *Python, данные, бит, код, алгоритм, функция, модуль, хэш, hashlib.*

WORK WITH DIGESTS MESSAGE IN PYTHON LANGUAGE

Otaxanov Nurillo Abdumalikovich,
professor of NamSU.

Annotation. *This article describes the process of processing various data using hash functions. It describes in detail the encryption algorithms used in working with hash functions. And also, examples of the practical application of the theoretical materials are shown.*

Keywords: *Python, data, bit, code, algorithm, function, module, hash, hashlib.*

Kriptografik xesh-funktsiya - ixtiyoriy o'lchamdagi ma'lumotlarni fiksirlangan o'lchamli bitlar massiviga aylantiruvchi matematik algoritmdir.

Xesh funksiyasi tomonidan ishlab chiqarilgan natija xesh summasi yoki oddiygina qilib xesh, kiruvchi ma'lumotlar esa ko'pincha xabar deb ataladi.

Ideal xesh funksiyasi uchun quyidagi shartlar bajariladi:

a) xesh funksiyasi bir xil xabar uchun bir xil xesh qiymatiga olib keladi;

b) xesh qiymati har qanday xabar uchun tezda hisoblanadi

c) berilgan xesh qiymatini hosil qiluvchi xabarning topib bo'lmasligi;

d) bir xil xesh qiymatiga ega bo'lgan ikki xil xabar mavjud emas;

e) xabardagi kichik o'zgarish xeshni shunchalik o'zgartiradiki, yangi va eski qiymatlar o'zaro bog'liq emasdek tuyuladi.

Python tili ma'lumotlar havfsizligi ta'minlash uchun xizmat qiladigan hashlib modulini o'z ichiga oladi. Bu modul ma'lumot va xabarlarni havfsiz xeshlash uchun zarur bo'lgan ko'plab algoritmlar uchun umumiy interfeysga ega. Bu interfeys o'z ichiga FIPS SHA1,

SHA224, SHA256, SHA384, SHA512 hamda RSA MD5 kabi algoritmlarni oladi. Avvallari xabarlar dayjesti deb atalgan tushuncha zamonaviy kipyografitada havfsiz xesh den ataladi.

Har bir tipdagi xeshlar uchun konstruktorning bitta metodi xizmat qiladi. Ularning barchasi bir xildagi sodda interfeysli xesh-obyektni qaytaradi. Hashlib modulida *sha1(), sha224(), sha256(), sha384(), sha512(), blake2b(), blake2s()* va boshqa konstruktorlar mavjud. Bu konstruktorlar bir-biridan shifrlash algoritmlari bilan farqlanadi. Masalan, "Python tili ma'lumotlar havfsizligini ta'minlaydi." matni uchun xabarlar dayjesti tanlangan konstruktorga ko‘ra quyidagicha bo‘lishi mumkin:

```
import hashlib
m = hashlib.sha256()
m.update(b"Python tili ma'lumotlar havfsizligini ta'minlaydi")
print('sha256 : ', m.hexdigest())
print('Xabar hajmi=', m.digest_size)
print('Blok hajmi=', m.block_size)
m = hashlib.sha224()
m.update(b"Python tili ma'lumotlar havfsizligini ta'minlaydi")
print('sha384 :', m.hexdigest())
print('Xabar hajmi=', m.digest_size)
print('Blok hajmi=', m.block_size)
m = hashlib.sha384()
m.update(b"Python tili ma'lumotlar havfsizligini ta'minlaydi")
print('sha224 :', m.hexdigest())
print('Xabar hajmi=', m.digest_size)
print('Blok hajmi=', m.block_size)
```

```
sha256 :  c4435aba0f8c00f47bcebf0ab1eb6187039d5633971281c6ec9dd0c73e26bff8
Xabar hajmi= 32
Blok hajmi= 64
sha384 : 9bb21fdceb88001a70d9b09e4f8099533b3e62f0220fefda3eeb3382
Xabar hajmi= 28
Blok hajmi= 64
sha224 : 0a0fec6ffafd2877d669aa18ff3724d9916a992a0c38e3beecc859781aefdc57091ad46bb86
10c0cfd3a1eff663db1dc
Xabar hajmi= 48
Blok hajmi= 128
>>>
```

Hashlib ning barcha konstruktorlari havfsizlikni ta’minlash uchun foydalaniladigan va

```
sha256 :  c4435aba0f8c00f47bcebf0ab1eb6187039d5633971281c6ec9dd0c73e26bff8
32
64
sha384 : 9bb21fdceb88001a70d9b09e4f8099533b3e62f0220fefda3eeb3382
28
64
sha224 : 0a0fec6ffafd2877d669aa18ff3724d9916a992a0c38e3beecc859781aefdc57091ad46bb86
10c0cfd3a1eff663db1dc
Xabar hajmi= 48
Blok hajmi= 128
>>>
```

to‘g‘ridan-to‘g‘ri *True* qiymatiga ega bo‘lgan argumentlarni qabul qiladi. *False* qiymati havfsiz bo‘lmagan xeshlash algoritmlariga nisbatan qo‘llanadi.

Hashlib moduli algoritmlaidan birini tanlash uchun New metodidan foydalaniladi. Buyruqning umumiy ko‘rinishi quyidagicha:

hashlib.new(name, [data,], usedforsecurity=True).*

Bu umumiy konstruktor bo‘lib, birinchi parametri sifatida algoritm nomi ko‘rsatilad Masalan,

h = hashlib.new('sha512_256').

Python tili quyidagi attributlarga ega:

hashlib.algorithms_guaranteed – mazkur modul tomonidan barcha platformalarda ishlash kafolatlangan xeshlash algoritmlari nomlaridan iborat ro‘yhat. Bu metod Pythonning 3.9 vesiyasi uchu quyidagi natijani beradi:

```
import hashlib
dk = hashlib.algorithms_guaranteed
print(dk)
```

```
>>> {'sha3_224', 'blake2s', 'sha3_512', 'sha3_384', 'sha3_256', 'shake_128', 'sha384'
, 'sha224', 'shake_256', 'sha256', 'sha1', 'sha512', 'md5', 'blake2b'}
>>>
```

hashlib.algorithms_available – ishlayotgan Pithon interpretatorida ochiq bo‘lgan xeshlas algoritmlari ro‘yhatini o‘z ichiga oladi. Bu ro‘yhat *new(). algorithms_guaranteed* uzatadigan qiymat doim qism to‘plam tarzida qabul qiladi.

```
import hashlib
dk = hashlib.algorithms_available
print(dk)
```

```
>>> {'sha3_512', 'sha256', 'md5-sha1', 'sha3_384', 'shake_256', 'blake2b', 'sha512_2
56', 'shake_128', 'md4', 'sha512', 'sm3', 'md5', 'whirlpool', 'sha3_224', 'sha512_
224', 'sha224', 'mdc2', 'ripemd160', 'blake2s', 'sha1', 'sha3_256', 'sha384'}
>>>
```

Konstruktorlar qaytaradigan xesh-obyektlarning doiniy attributlari quyidagilarda iborat:

hash.digest_size	natijaviy xeshning baytlardagi hajmi
hash.digest_size	xeshlash algoritmlari ichki blogining baytlardagi hajmi

Xesh-obyektlar *hash.name* (joriy xeshning kanonik nomi) hususiyatiga ega. Bunda tashqari, xesh-obyektlar uchun quyidagi metodlar xizmat qiladi:

hash.update(ma'lumot)	xesh-obyektni bitlardagi obyekt bilan yangilash. Metodga takroriy murojaat birlashma amalini beradi. Masalan, *m.update(a); m.update(b)* buyruqlari *m.update(a+b)* ga ekvivalent.
hash.digest()	joriy vaqtda *update()* metodiga uzatilgan ma'lumotlar dayjestini qaytaradi. Bunda uning hajmi 0 dan 255 gacha bo‘lishi mumkin.
hash.hexdigest()	*Digest()* kabi ishlaydi. Faqat ma'lumotlar 16 lik sanoq sistemasida ifodalanishi bilan farqlanadi.
hash.copy()	xesh-obyekt nisxasini (klonini) qaytaradi. Odatda umumiy ostsatrga ega bo‘lgan ma'lumot dayjestlarini qayta ishlashda qo‘llanadi.

Uzunligi o‘zgaruvchan bo‘lgan SHAKE dayjestlari. ***Shake_128()*** **va** ***shake_256()*** **algoritmlari uzunligi havfsizlik biti** ***length_in_bits*** **// 2 dan 128 yoki 256 gacha bo‘lgan dayjestlarni ishlab chiqishga yordam beradi. Shunday qilib, bu metodlar dayjest uzunligini talab qiladi. Maksimal uzunlik SHAKE algoritmida chegaralanmagan.**

Shake.digest(length) – bu metod joriy vaqtda *update()* ga uzatilgan dayjestlrni qaytaradi. Bu baytli obyektning uzunligi 0 dan 255 gacha bo‘lishi mumkin.

shake.hexdigest(length) – huddi *shake.digest()* kabi faqat 16 lik sanoq sistemasida) ishlaydi. Bu metod odatda elektron pochta va boshqa nobinar muhitlarda qo‘llanadi.

Kalitlarni himoyalash (parollash) ning yaxshi funksiyasi sozlanuvchan va o‘z ichiga *salt* ni olishi kerak. Bunday funksiyanng umumiy ko‘rinishi quyidagicha:

hashlib.pbkdf2_hmac(hash_name, password, salt, iterations, dklen=None).

Bu yerda *hash_name* – kalit uchun xesh-dayjest algoritmining nomi; *password* va salt batli bufferlar tarzida talqin qilinadi. *Password* ning uzunligi 1024 gacha, salt niki esa 16 va undan katta bo‘lishi mumkin. Iteratsiyalar soni xeshlash algoritmiga ko‘ra tanlanadi. 2013 yildagi holatga ko‘ra SHA256 algoritmida ular 100000 dan kam bo‘lmasligi lozim; *dklen* – hosila kalitning uzunligi. Agar u None bo‘lsa, u holda *hash_name* algoritmidagi dayjest hajmidan foydalaniladi (masalan, sha512 uchun 64 ga teng).

```
import hashlib
dk = hashlib.pbkdf2_hmac('sha256', b'password', b'salt', 100000)
print(dk.hex())
```

```
0394a2ede332c9a13eb82e9b24631604c31df978b4e2f0fbd2c549944f9d79a5
>>>
```

Quyidagi funksiya kalitlarni scrypt paroli asosida aniqlashni ta’minlaydi:

*hashlib.scrypt(password, *, salt, n, r, p, maxmem=0, dklen=64).*

Masalan:

```
import hashlib
dk = hashlib.scrypt(b'password', salt=b'salt', n=2, r=8, p=1)
print(dk)
```

```
b'cm\x89\x85\xf1\x14\x8f\x8a\x10\xf9\xf9%\xf4\xe3\xe8\x95\xb8g\xbd\xf4:\x8fy
o\xc8\xc4\x99&@e\x19\xfa\xe4\xa2\x9b.I/v\xce;\x0b\xd9aC&K\x04\xee\x86\xd
e\xcf\x16\xf9\xc19mM\xe9n\xa4S\xb8\xa2'
>>>
```

Bu yerda password va salt baytli obyektlar bo‘lib, password ning uzunligi imkon doirasida chegaralangan (masalan, 1024), salt esa 16 yoki undan kattaroq bo‘lishi lozim; dklen – hosila kalitning uzunligi.

BLAKE2- bu RFC 7683 da (xesh va xabar autentifikatsiyalash standarti) belgilangan kriptografik xesh-funksiya bo‘lib, ikki hil ko‘rinishda qo’llanadi:

- **BLAKE2b** – 64 bitli platformalar uchun moslashtirilgan va 1 dan 64 bauygacha bo‘lgan dayjestlarni yaratish uchun xizmat qiladi;
- **BLAKE2s** – 8 va 32 bitli platformalar uchun optimallashtirilgan va 1 dan 32 baytgacha bo‘lgan dayjestlarni yaratish uchun xizmat qiladi .

BLAKE2 kalitli rejimda, salt yordamida xeshlashtirish va individuallashtirish asosiga ishlaydi.

Yangi xesh-obyektlar quyidagi konstruktorlardan biri asosida yaratiladi:

a) *hashlib.blake2b(data=b'', *, digest_size=64, key=b'', salt=b'', person=b'', fanout=1, depth=1, leaf_size=0, node_offset=0, node_depth=0, inner_size=0, last_node=False, usedforsecurity=True)*

b) *hashlib.blake2s(data=b'', *, digest_size=32, key=b'', salt=b'', person=b'', fanout=1, depth=1, leaf_size=0, node_offset=0, node_depth=0, inner_size=0, last_node=False, usedforsecurity=True).*

Bu funksiyalar BLAKE2b yoki BLAKE2s larni hisoblash uchun xesh-obyektlarni qaytaradi. Uning parametrlari quyidagi ma'nolarni anglatadi:

- *data* – baytli obyekt, xeshlash uchun boshlang'ich ma'lumotlar parchasi;
- *digest_size* -chiquvchi dayjestning baytlardag hajmi ;
- *key* – xeshlash uchun kalit;
- *salt* – tasodifiylashtirilgan xeshlash uchun tuz;
- *person* – individuallashtirish satri.

Quyidagi jadvalda umumiy parametrlarning eng yuqori chegaralari baytlarda keltirilgan:

Hash	digest_size	len(key)	len(salt)	len(person)
BLAKE2b	64	64	16	16
BLAKE2s	32	32	8	8

Shuningdek, konstruktor funksiyasi xeshlsh daraxtining quyidagi parametrlarini qabul qilishi mumkin:

- *fanout* - fanout (0 … 255, 0 – agar cheklanmagan bo'lsa, 1 – ketma-ketlik rejimida;
- *depth* – daraxtning maksimal chuqurligi (1 … 255, 255 – agar cheklanmagan bo'lsa, 1 – ketma-ketlik rejimida);
- *leaf_size* – yaproqning baytlardagi maksimal uzunligi (0 … 2^{32}-1, 0 … 255, 0 – agar cheklanmagan yoki ketma-ketlik rejimida bo'lsa)';
- *node_offset* – tugunning surilishi (BLAKE2b uchun 0 … 2^{32}-1, BLAKE2 uchun 0 … 2^{48}-1);
- *node_depth* – tugunning chuqurligi (0 …255, 0 – yaproqlar uchun);
- *inner_size* – ichki dayjest hajmi (BLAKE2b uchun 0 … 64, BLAKE2 uchun 0 … 32;
- *last_node* – qayta ishlangan tugunning oxirgisimi yoki yo'qligini ko'rsativchi mantiqiy parametr (ketma-ketlik rejimi uchun -False).

BLAKE2b va BLAKE2s lar uchun quyidagi konstantalardan foydalaniladi:

.SALT_SIZE	salt ning maksimal uzunligi
.PERSON_SIZE	individuallashtirish satri hajmi
.MAX_KEY_SIZE	kalitning maksimal hajmi
.MAX_DIGEST_SIZE	xesh-funksiya qaytaradigam dayjestning maksimal hajmi

Quyidagi namunada dastlab funksiya-konstruktor (BLAKE2d yoki BLAKE2s) yordamida xesh-obyekt yaratilmoqda. So'ngra undagi ma'lumotlar update() asosida yangilanib, ekranga uzatilmoqda.

```
from hashlib import blake2b
h = blake2b()
h.update(b'Salom Python')
print(h.hexdigest())
```

```
46f579767ac09b450307bde7bcfb9522c024f6844320b5478b7434bf8b35a5cbbd
5ec17ff1b5593dd94a9b29abfc7792ebcac20ce80e8017cb3dfb01eccb8c01
>>>
```

BLAKE2 dayjestlar uchun turli uzunliklarda (BLAKE2b uchun 64 baytgaca, BLAKE2s uchun 32 baytgacha) bo‘lishi mumkin.

```
from hashlib import blake2b
h = blake2b(digest_size=30)
h.update(b'Salom Python')
print(h.hexdigest())
```

```
ef400321e2a23cadc423164f9bc006a43bfcc2d69b2a282e0aac376e469e
>>>
```

Turli hajmdagi xeshlash obyektlarining chquvchi dayjestlari turlicha bo‘ladi:

```
from hashlib import blake2b
print(blake2b(digest_size=10).hexdigest())
print(blake2b(digest_size=11).hexdigest())
```

```
6fa1d8fcfd719046d762
eb6ec15daf9546254f0809
>>>
```

Foydalanilgan adabiyotlar ro‘yhati

1. hashlib - безопасные хэши и дайджесты сообщений. https://runebook.dev/ru/docs/python/library/hashlib

ВЕРОЯТНОСТЬ НАСЛЕДОВАНИЯ НЕСВЯЗАННЫХ ГЕНОВ В 5-М ПОКОЛЕНИИ

Полванов Р.Р., Шарипов Ф.М. (НамГУ)

Аннотация: *Работа посвящена расчетам вероятностей для 4 пар несвязанных генов в 5-м поколении.*

Ключевые слова: *доминантный, рецессивный, несвязанный, гаметы, хромосомы, скрещивания, панмиксия.*

THE PROBABILITY OF INHERITANCE OF NON-LINKED GENES IN THE 5TH GENERATIONS

R.R. Polvanov, F. M. Sharipov (NamSU)

Abstract: *The work is devoted to the probability calculations for 4 pairs of uncoupled genes in the 5th generation*

Key words: *Dominant, recessive, uncoupled, gametes, chromosomes, crosses, panmixia.*

As you know, the phenomenon of suppression of one characteristic by others is natural. *A* sign that suppresses another is called dominant, and the next is recessive. Dominant denote by the capital letter *A*. Recessive sign small letter *a*. Each gene has two varieties - dominant and recessive trait. Therefore, genes go in pairs.

МУНДАРИЖА

ФИЗИКА-МАТЕМАТИКА ФАНЛАРИ
01.00.00 ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ
PHYSICAL AND MATHEMATICAL SCIENCES

1 **Yorug`lik nurining yuqori sezgir bo`yoq tarkibli quyosh foto elementlariga ta'sirini o`rganish.**
To`lqinov M.A. 3

2 **A-Si:H асосидаги структураларда ёруғликни нотекис ютилишини фотоэлектрик параметрларга таъсирини лазер ёрдамида тадқиқ қилиш.**
Бабаходжаев У.С., Набиев А.Б., Нематуллаев Ж.Р., Исабоева Ф.Д., Хайдарова Ф.Б., Тўхтаралиев А.Ш. 7

3 **Python tilida xabar Dayjestlari bilan ishlash**
Otaxanov N.A. 13

4 **Вероятность наследования несвязанных генов В 5-М поколении**
Полванов Р.Р., Шарипов Ф.М. 18

5 **Иммиграцияли Беллман-Харрис тармоқланиш жараёнини яшаш даври**
Машраббоев А., Умматалиев У.И., Ибрагимова Н.А. 20

КИМЁ ФАНЛАРИ
02.00.00 ХИМИЧЕСКИЕ НАУКИ
CHEMICAL SCIENCES

6 **Metallarni cho'zish uchun olingan yangi compositini infraqizil spektrofotometr tahlili**
Doliev G'., Abdulkayev A., Umaraliev J., Jo'raev B., G'ofurov I. 24

7 **Shaftoli mevasining kimyoviy tarkibi va inson organizmiga ta'siri.**
Dehhqonov R.S., Muminova M.R. 29

8 **Phlomoides Kaufmanniana o'simligining element tahlili.**
Muradov M.T., Karimov A.M. 33

9 **Murakkab oksidli birikmalarda piroxlor tipli tuzilishga ega $Na_xK_{Y-x}SB_YW_{2-Y}O_6$ tarkibli fazalar hosil bo'lishi**
Bozorov X.N., Lupitskaya Yu.A., Doliyev G.A., Buchelnikov V.D., Abdullaeva G.U. 37

10 **Fatalimid asosida olingan sorbentning sorbsion sig'imini aniqlash**
G'afforova Sh., Turayev H.X., Sottiqulov E.S., Babamuratov B.E. 41

11 **Metallarni cho'zish uchun olingan yangi compositini elektron mikroskop va element tahlili tahlili**
Doliev G'. A., Abdulkayev A. B., Umaraliev J., Xabibullaev X., Saydullaea G. 46

12 **Карбоксиметилхитозан Bombyx Mori асосида нанотола олиш шароитлари**
Саттарова Д.М., Саттаров Т.А. 51