საქართველოს უნივერსიტეტი

მეცნიერებისა და ტექნოლოგიების სკოლა

ხელნაწერის უფლებით

არტურო არაქელიანი

პოსტ-კვანტური ციფრული ხელმოწერის ასინქრონული ალგორითმი

მეცნიერებისა და ტექნოლოგების სკოლის

დოქტორის აკადემიური ხარისხის მოსაპოვებლად წარმოდგენილი ნაშრომის

სადისერტაციო მაცნე

თბილისი

2022

სადისერტაციო ნაშრომი შესრულებულია საქართველოს უნივერსიტეტის მეცნიერებისა და ტექნოლოგების სკოლაში.

სამეცნიერო ხელმძღვანელები: **მაქსიმ იავიჩი, ბექარ მელაძე**

გარე ექსპერტები: **გიორგი იაშვილი, ელზა ჯინჭარაძე, ლელა მირცხულავა**

დისერტაციის დაცვა შედგება .
მისამართი: თბილისი, საქართველოს უნივერსიტეტი, კოსტავას 77ა, #519 აუდიტორია.
დისერტაციის გაცნობა შეიძლება საქართველოს უნივერსიტეტის ბიბლიოთეკაში

სადისერტაციო მაცნე დაიგზავნა .

სადისერტაციო საბჭოს მდივანი: **ნათია მანჯიკაშვილი**

შესავალი

თემის აქტუალურობა. ბოლო დროს სულ უფრო აქტუალური ხდება კვანტური კომპიუტერები. შესაბამისად არსებული სისტემები, რომელიც დაფუძნებულია მარტივი რიცხვების ფაქტორიზაციზე, ხდება არაუსაფრთხო და დაუცველი. ასეთი სისტემების თვალსაჩინო მაგალითი არის RSA . შესაბამისად დგება უსაფრთხო სისტემის შექმნის ან არსებული სისტემების დახვეწის საკითხი. არსებობს მრავალი ხელმოწერის სისტემა, ერთჯერადი და არა მხოლოდ.

კრიპტოგრაფიის ფუნდამენტური მიზანი არის კონფიდენციალურობის, მონაცემების მთლიანობის, უტყუარობის, აუტენთიფიკაციის და საიმედოობის უზრუნველყოფა. სწორედ კრიპტოგრაფიის დახმარებით ხდება თაღლითობის, არასანქცირებული წვდომისა და სხვა მსგავსი დანაშაულებრივი ქმედებების აღმოჩენა და პრევენცია.

ნაშრომში განხილულია კრიტოგრაფიული მეთოდების, სტანდარტები, მათი თეორიული და პრაქტიკული გამოყენების მნიშვნელობა. ასევე წარმოდგენილია კრიპტოგრაფიული მეთოდების სხვადასხვა ალგორითმი.

შესწავლილია ლიტერატურაში მოყვანილი სტატიები და განხილულია მათი შედეგები. სტატიებიდან ნათლად ჩანს, რატომ არის მნიშვნელოვანი ახალი ალგორითმებისა და მათთვის სათანადო სიგრძის გასაღების შერჩევა. სტატიებში ასევე გაანალიზებულია ცნობილი თანამედროვე კრიპტოსისტემების შეტევები და სუსტი წერტილები.

2016 წელს გამოქვეყნდა სტატია იმის შესახებ, რომ კორპორაცია Google-მა, NASA-მ და კოსმოსური კვლევების უნივერსიტეტების ასოციაციამ (Universities Space Research Association-USRA) მოაწერეს ხელი თანამშრომლობაზე კვანტური D-Wave პროცერორების მწარმოებელთან.

D-Wave 2X უახლესი კვანტური პროცესორია, რომელიც შეიცავს 2048 ფიზიკურ კუბიტს (კვანტური განმუხტვები, ინფორმაციის შენახვის უმცირესი ერთეულები კვანტურ კომპიუტერში). 1152 კუბიტი კვანტური კომპიუტერის ამ მოდელში გამოიყენება

გამოთვლების შესასრულებლად. თითოეული დამატებითი კუბიტი ორჯერ ზრდის ძიების სივრცეს, შესაბამისად იზრდება გამოთვლების სიჩქარეც.

ჰეშ ფუნქციები, რომელიც აქტიურად გამოიყენება სხვადასხვა ხელმოწერის სისტემებში, როგორიცაა Merkle-ს ხელმოწერის სისტემა, ასევე აქტიურად გამოიყენება სხვადასხვა მონაცემის დასაჰეშად მონაცემთა ბაზაში. ილუსტრაცია 2-ზე ნაჩვენებია პაროლის ველი მონაცემთა ბაზაში (MySQL), სადაც bcrypt-ის დახმარებით დაჰეშილი პაროლი ინახება.

თანამედროვე პროცესორების განვითარება წინ მიდის, გამოდის უფრო ახალი, ეფექტური და მრავალნაკადიანი პროცესორები. არსებობს ბევრი, ჰეშზე დაფუძნებული ალგორითმი, რომელიც არის მდგრადი კვანტური კომპიუტერების შეტევებისგან, მაგრამ არის არაეფექტური.

სადოქტორო დისერტაციის მიზანი არის ისეთი სისტემის წარმოდგენა, რომელიც იქნება მდგრადი კვანტური კომპიუტერების შეტევებისგან და ამავდროულად საკმაოდ ეფექტური.

მეცნიერული სიახლე. სადისერტაციო ნაშრომის მეცნიერული სიახლეს წარმოადგენს არსებული პოსტ-კვანტური კრიპტოსისტემის ახალი ალგორითმი, რომლის დახმარებით სისტემა არის არა მხოლოდ უსაფრთხო, არამედ ბევრად სწრაფი და ეფექტური. ჩვენს მიერ წარმოდგენილი ალგორითმი იყენებს პროცესორის ნაკადებს შიფრაციის გასაღების გამოსათვლელად.

გასაღების გენერაცია: ზომა უნდა იყოს H >= 2 იმისათვის, რომ მოხდეს ერთი public key - ით 2H დოკუმენტის ხელმოწერა. აქვე ხდება ხელმოწერისა და დადასტურების გასაღებების გენერაცია: Xi, Yi, 0<=i<=2H . Xi არის ხელმოწერის გასაღები, ხოლო Yi არის დადასტურების გასაღები.

იმისათვის, რომ მივიღოთ მშობელი კვანძი, საჭიროა გავაერთიანოთ 2 წინა კვანძი(შვილი) და მოვახდინოთ ჰეშირება; a[i, j] არის ხის კვანძები;

$$a[1,0]=h(a[0,0] \,||\, a[0,1]) .$$

ხის გაყოფა ხდება პროცესორის ნაკადების რაოდენობაზე. ციკლში, რომლის სიგრძე ტოლია ნაკადების რაოდენობის, ვახდენთ მშობელი კვანძების გამოთვლას. დავუშვათ, გვაქვს t რაოდენობის ნაკადი და d კვანძი. d კვანძების რაოდენობას ვყოფთ t ნაკადების რაოდენობაზე: d / t . d / t კვანძები ეშვება ცალკეულ ნაკადებში. მშობელი კვანძები მიიღება (40) გამოსახულების მიხედვით. ხდება მიღებული სიმრავლეების კონკატენაცია და შემდგომ ხდება მათი t სიმრავლეებად დაყოფა. ეს პროცესი გრძელდება მანამ, სანამ არ მივიღებთ ხის ფუძეს. ხის ფუძე არის public key .

შეტყობინების ხელმოწერა: იმისათვის, რომ მოვახდინოთ შეტყობინების ხელმოწერა, ჰეშირების საშუალებით ვახდენთ მის ტრანსფორმაციას n ზომამდე. h(m) = hash. იმისათვის, რომ მოხდეს შეტყობინების ხელმოწერა, საჭიროა ნებისმიერი ერთჯერადი Xany გასაღების, ერთჯერადი ხელმოწერის, ერთჯერადი verification გასაღებისა და ყველა მეზობელი კვანძების გამოყენება.

$$\text{Signature} = (\text{sig} \,||\, \text{any} \,||\, \text{Yany} \,||\, \text{auth0} ,\ldots, \text{authH-1}).$$

ხელმოწერის დადასტურება: იმისათვის, რომ მოხდეს ხელმოწერის დადასტურება საჭიროა ერთჯერადი ხელმოწერის შემოწმება verification გასაღების დახმარებით. იმ შემთხვევაში თუ განხორციელდება ხელმოწერის დადასტურება, მაშინ ხდება ყველა a[i, j] ელემენტის გამოთვლა auth , index , any , Yany გამოყენებით. თუ ხის ფუძე (root) ემთხვევა public-key - ს, მაშინ ხელმოწერა სწორია.

კვლევის მიზანი. სადისერტაციო ნაშრომში წარმოდგენილი კვლევის მიზანია ისეთი კრიპტოგრაფიული ალგორითმის წარმოდგენა, რომელიც მდგრადი და ეფექტური იქნება კვანტური კომპიუტერების მხრიდან მომდინარე პოტენციური შეტევების მიმართ.

მეცნიერები აქტიურად მუშაობენ კვანტური კომპიუტერების შექმნაზე. კვანტურ კომპიუტერებს შეეძლება დიდი ციფრების ფაქტორიზაციის განხორციელება. შესაბამისად, კვანტურ კომპიუტერებს შეეძლება RSA ალგორითმის გატეხვა, რომელსაც დღესდღეისობით

ბევრი პროგრამა იყენებს. ჰეშზე დაფუძნებული ციფრული ხელმოწერები არის RSA - ს ალტერნატივა. ეს სისტემები იყენებს ჰეშ ფუნცქიებს. ამ სისტემების დაცულობა დამოკიდებული არის ჰეშ ფუნქციების კოლიზიაზე.

Merkle-ს კლასიკური ალგორითმი შეიძლება მიჩნეულ იქნას როგორც სტატიკური, რადგან ის არ არის დამოკიდებული პროცესორის ნაკადების რაოდენობაზე. ჩვენ გთავაზობთ ალგორითმს, რომელიც იყენებს პროცესორის ნაკადებს. აქ არის წარმოდგენილი ამ ალგორითმის მათემატიკური მოდელი და ალგორითმის ფსევდო კოდი. მისი ეფექტურობა დასტურდება მიღებული შედეგებით.

რადგან კვანტური კომპიუტერების შემუშავება აქტიურ ფაზაშია და Shor - ის ალგორითმის დახმარებით შეუძლიათ მარტივად გატეხონ სისტემები, რომლებიც იყენებს რიცხვების ფაქტორიზაციას, ბევრი არსებული სისტემა ხდება დაუცველი.

კვლევის ობიექტები. კვლევის ობიექტები არის არსებული კლასიკური ალგორითმები და ასევე ჰეშზე დაფუძნებული შიფრაციის ალგორითმები.

კვანტური კომპიუტერის მიერ RSA, DSA და ECDSA კრიპტოგასაღების გატეხვის შემდეგ, ჩვეულებრივი ინტერნეტ მომხმარებლებისთვის შესაძლებელია შეიქმნას ისეთი წარმოდგენა, რომ თანამედროვე შეტევების მიმართ არსებული სისტემები დაუცველია. ინტერნეტ მომხმარებლებისთვის ან კომპანიებისთვის დაცვის ერთადერთი საიმედო გამოსავალი, რათა ინფორმაცია არ იყოს ხელმისაწვდომი ჰაკერებისთვის შეიძლება იყოს ფიზიკური ფარი. მაგალითად, USB მეხსიერების დამალვა კარგად დაცულ სეიფში. შეიძლება ვივარაუდოთ, რომ კვანტურ კომპიუტერებს შეუძლიათ RSA , DSA და ECDSA მარტივად გატეხვა ან ჩავთვალოთ, რომ კვანტური კომპიუტერები უსარგებლოს გახდის კრიპტოგრაფიას. RSA, DSA და ECDSA უკან დგას კრიპტოგრაფიის ბევრი და მნიშვნელოვანი კლასი:

1. ჰეშირებაზე დაფუძნებული კრიპტოგრაფია. კლასიკური მაგალითი არის Merkle- ს ჰეშ public-key - ზე დაფუძნებული სისტემა (1979), რომელიც არის დაფუძნებული Lamport - ის და Diffie - ის იდეაზე.
2. კოდზე დაფუძნებული კრიპტოგრაფია. კლასიკური მაგალითი არის McEliece’s - ის დამალული Goppa კოდი public-key - ზე დაფუძნებული სისტემა (1978).

3. Lattice - ზე დაფუძნებული კრიპტოგრაფია. მაგალითი, რომელმაც ალბათ ყველაზე დიდი ინტერესი გამოიწვია, არის Hoffstein-Pipher-Silverman NTRU public-key - ზე დაფუძნებული სისტემა (1998).
4. Secret-key კრიპტოგრაფია. ყველაზე კარგი მაგალითი არის Daemen-Rijmen Rijndael შიფრი(1998), რომელსაც შემოკლებით ჰქვია AES (Advanced Encryption Standard) .

მიჩნეული არის, რომ ეს დაცვის სისტემები უძლებს კლასიკურ და კვანტურ კომპიუტერებს. ჯერ-ჯეროობით ვერავინ ვერ გადაწყვიტა, როგორ მოარგონ Shor - ის ალგორითმი, რომელიც ადვილად უმკლავდება RSA - ს, DSA - ს და ECDSA - ს რომელიმე ზემოთ აღწერილ სისტემას. სხვა კვანტური ალგორითმი არის Grover - ის ალგორითმი. ის არ არის ისეთი სწრაფი, როგორიც არის Shor - ის ალგორითმი.

პრაქტიკული მნიშვნელობა. დისერტაციის ფარგლებში შემუშავებულია კროიპტოგრაფიული ალგორითმი, რომლის გამოყენება შესაძლებელია პრაქტიკაში. დისერტაციაში მიღებული შედეგების ანალიზის საფუძველზე დგინდება რომ, ჩვენს მიერ შემუშავებული ალგორითმის გამოთვლის სიჩქარე, რომელიც ეფუძნება Merkle-ს კლასიკურ ალგორითმს, არის რამდენჯერმე სწრაფი სხვა არსებულ ალგორითმებთან შედარებით.

კვლევის ძირითადი შედეგები. წარმოდგენილი ალგორითმი საშუალებას იძლევა სრულად იქნას გამოყენებული თანამედროვე პროცესორის რესურსები, პროცესორის ნაკადებზე დაყრდნობით უზრუნველყოფილი იქნას პარალელური გამოთვლები, რის შედეგადაც ხორციელდება ოპერაციების რაოდენობისა და გამოთვლების საწარმოებლად საჭირო დროის მნიშვნელოვნად შემცირება.

დისერტაციის ფარგლებში შესრულდა შემდეგი სახის ამოცანები:

1. კრიპტოგრაფიის არსებული სიმეტრიული და ასიმეტრიული ალგორითმების ანალიზი.

2. დღესდღეისობით არსებული კლასიკური (ფართოდ გამოყენებადი) კრიპტოგრაფიის ალგორითმების პოსტ-კვანტურ ალგორითმებთან შედარება.

3. ნაშრომში განხილულია ახალი ალგორითმის აგების პროცესი, რომელიც ეფუძნება Merkle-ს ხეს და ამავდროულად წარმოადგენს პარალელურ ალგორითმს.

4. განხორციელდა მიღებული ალგორითმის ეფექტურობის ანალიზი და მისი სისწრაფის (ოპერაციების რაოდენობის) შედარება არსებულ ალგორითმებთან.

შემუშავებული ალგორითმის ეფექტურობა დასტურდება ექსპერიმენტების შედეგებით. კერძოდ, 8 ნაკადიანი პროცესორის შემთხვევაში ოპერაციების რაოდენობა შემცირდა 2.3 ჯერ, ხოლო 16 ნაკადიანი პროცესორის შემთხვევაში ოპერაციების რაოდენობა შემცირდა 9 ჯერ. აღნიშნული ალგორითმის დახმარებით მნიშვნელოვნად შემცირდება შიფრაცია/ვერიფიკაციის დროც. ექსპერიმენტების პროცესში, გამოთვლების წარმოებისას 2 ნაკადიან პროცესორზე ახალი ალგორითმი შესრულდა 3,57 ჯერ უფრო სწრაფად ვიდრე კლასიკური. პრაქტიკაში გვხვდება უფრო მეტი ნაკადის მქონე პროცესორები, სადაც ჩვენს მიერ მიღებული ალგორითმის შესრულების სიჩქარე იქნება კიდევ უფრო სწრაფი.

დისერტაციის სტრუქტურა და მოცულობა. ნაშრომი შედგება: აბსტრაქტისგან, მიმოხილვისგან, 3 თავისგან, 37 ქვეთავისგან, დასკვნისგან და გამოყენებული ლიტერატურის სიისგან. ნაშრომში წარმოდგენილია 14 გრაფიკული გამოსახულება.

მერკლეს კლასიკური ალგორითმის, ფრაქტალური ალგორითმის და " threads " - ებზე დაფუძნებული ალგორითმის ოპერაციების დათვლა. *კლასიკური ალგორითმის საშუალო ხარჯები(Average costs).* ხეში ყოველი კვანძი საბოლოო ჯამში არის authentication path - ის ნაწილი. დავუშვათ, გვაქვს 2H-h მარჯვენა(შესაბამისად, მარცხენა) კვანძები h სიმაღლის ხეში. თუ დავითვლით ყოველ კვანძს დამოუკიდებლად, ყველა მათგანის საფასური იქნება 2h+1 - 1 ოპერაცია. რომ შევაჯამოთ გვექნება 2H+1 = 2N ოპერაცია. ყოველი h ($0 \le h \le H$) სიმაღლისთვის, ყველა საფასურის დაჯამებისას, საშუალოდ გვექნება $2H = 2\log(N)$ სავალდებულო ოპერაცია.

ალგორითმის შედეგის გამოთვლის ფაზა შედგება N ციკლისგან, თითო ფოთლისთვის გვაქვს $s \in \{0,\dots,N-1\}$. ყოველი s ციკლის დროს, authentication path-ი s - ური ფოთლისთვის შედეგი არის 〖AUTH〗_i,i=0,…,H-1.

ფრაქტალური ალგორითმის დრო. იქედან გამომდინარე, რომ გვაქვს R - 1 რაოდენობის ხე, საერთო გამოთვლითი საფასური ერთი ციკლისთვის იქნება:

$$T_{max} = 2(R-1) < 2S/s.$$

1. Set l = 0.
2. Output Authentication Path for leaf number l.
3. Next Subtree For each $\boldsymbol{j} \in \{\mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{S}\}$ for which $EXISTS_j$ is no longer needed, i.e, $l = 0 \pmod{2^{sj}}$:
    a. Remove Pebbles in $EXIST_j$.
    b. Rename tree $DESIRE_j$ as tree $EXIST_j$.
    c. Create new, empty tree $DESIRE_j$ (if $l + 2^{sj} < 2^S$).
4. Grow Subtrees For each $\boldsymbol{j} \in \{\mathbf{1}, \mathbf{2}, \ldots, \boldsymbol{s}\}$: Grow tree $DESIRE_j$ by applying 2 units to the modified treehash algorithm (unless $DESIRE_j$ is completed).
5. Increment l and loop back to step 2 (while $l < 2^H$).

ფრაქტალური ალგორითმის სივრცე(Space). აღნიშნული ალგორითმისთვის საჭირო დისკური მეხსიერების ზომა შეგვიძლია დავადგინოთ არსებული ქვეხეების, desired ქვეხეების და ე.წ. tails - ების დახმარებით.

ვთქვათ მოცემული გვაქვს R არსებული ქვეხეები და R - 1 desired ქვეხეები და ყოველი მათგანი შედგება 2s+1-2 pebbles - გან. დამატებით tail - თან ასოცირებული desired ქვეხე j > 1 დონეზე(level) შეიცავს s ·j+1 pebbles - ებს. აქედან გამომდინარე, გვაქვს:

$$SPACE_{max} \leq (2R - 1)(2^{s+1} - 2) + R - 2 + s(R - 2)(R - 1)/2.$$

ყველაზე ცუდი შედეგისათვის გვექნება:

$$SPACE_{max} < 2\ R\ 2^{s+1} + S\ R/2.$$

ნაკადებზე დაფუძნებული ალგორითმის ოპერაციების რაოდენობა. ახლა დავითვალოთ მიმდევრობითი ოპერაციების რაოდენობა ახალ ალგორითმში და ასევე შევადაროთ ის კლასიკურ ალგორითმს. ქვემოთ არის მოყვანილი ცხრილები, სადაც ნაჩვენებია ოპერაციების რაოდენობა კლასიკური და ახალი ალგორითმებისთვის, სხვადასხვა რაოდენობის ნაკადების მიხედვით.

ახალი ალგორითმი 8 “ thread ” - იან პროცესორში	
კვანძების რაოდენობა	მიმდევრობითი ოპერაციების რაოდენობა
8	3
64	10
1024	130
4096	514
16384	2050
262144	30722
1048576	129026

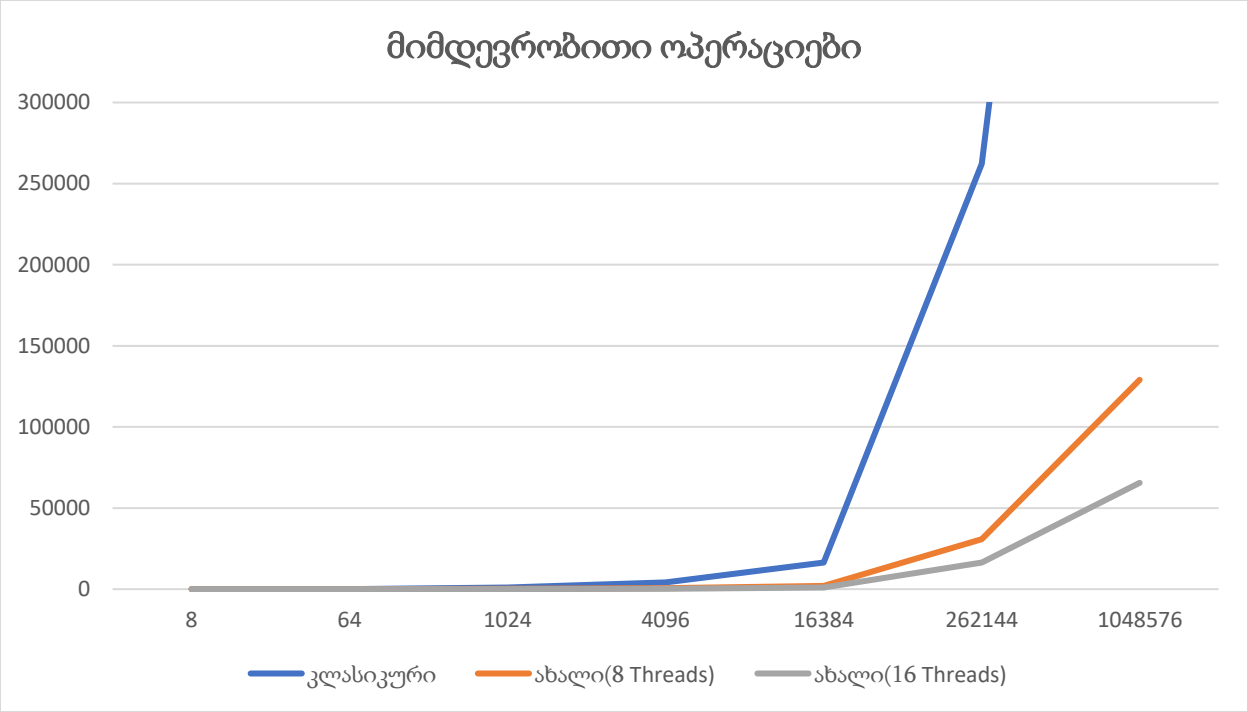
ახალი ალგორითმი 16 “ thread ” - იან პროცესორში	
კვანძების რაოდენობა	მიმდევრობითი ოპერაციების რაოდენობა
8	3
64	7
1024	67

4096	259
16384	1027
262144	16387
1048576	65539

ძველი ალგორითმი	
კვანძების რაოდენობა	მიმდევრობითი ოპერაციების რაოდენობა
8	7
64	63
1024	1023
4096	4095
16384	16383
262144	262143
1048576	1048575

აღნიშნულ მონაცემებზე დაყრდნობით შეგვიძლია მივიღოთ ახალი ალგორითმისთვის მიმდევრობითი ოპერაციების რაოდენობის გამოსათვლელი ფორმულა. დავუშვათ t არის ნაკადების რაოდენობა და n არის კვანძების რაოდენობა, გამოვთვალოთ O :

$O = n / t + \log_2 t/2$, თუ $n = t$, მაშინ $t = t / 2$


მიმდევრობითი ოპერაციები
300000
250000
200000
150000
100000
50000
0
8
64
1024
4096
16384
262144
1048576
კლასიკური
ახალი(8 Threads)
ახალი(16 Threads)

დასკვნა

ანალიზი. ჩვენს მიერ მიღებული ალგორითმი შევადარეთ კლასიკურ ალგორითმს. ტესტი ჩატარებული იქნა ისეთ კომპიუტერზე, რომლის პროცესორსაც გააჩნდა 2 ნაკადი. შეტყობინების სიგრძე კი იყო 128 ბიტი.

კლასიკური ალგორითმის შედეგები:

1. გასაღების გენერაციის დრო: 0.049351 წამი
2. ხელმოწერის დრო: 0.0002425 წამი
3. დადასტურების დრო: 0.0038651 წამი

Threads - ებზე დაფუძნებული ალგორითმის შედეგები:

1. გასაღების გენერაციის დრო: 0.013841 წამი
2. ხელმოწერის დრო: 0.0002425 წამი
3. დადასტურების დრო: 0.0038651 წამი

ამ ექსპერიმენტიდან გამომდინარე, შეგვიძლია ვნახოთ, რომ წარმოდგენილი ალგორითმი კლასიკურ ალგორითმზე არის 3,57- ჯერ სწრაფი.

ანალიზის შედეგად, შეგვიძლია დავასკვნათ, რომ ჩვენ მიერ შემუშავებული ალგორითმი გვაძლევს კარგ აჩქარებას გასაღების გენერაციის დროს (სხვა არსებულ ალგორითმებთან შედარებით).

დღევანდელ დღეს წარმოდგენილი ალგორითმი არის აქტუალური, რადგან დღევანდელი პროცესორები არის ძალიან სწრაფი და დინამიური. მათი განვითარება ხდება ძალიან სწრაფი ტემპით. შესაბამისად, ეს ალგორითმიც არის დინამიური. მისი სისწრაფე დამოკიდებულია პროცესორის სისწრაფესა და ნაკადების რაოდენობაზე.

სადისერტაციო ნაშრომის ძირითადი დებულებები გამოქვეყნებულია შემდეგ პუბლიკაციებში:

1. Iavich M., Arakeliani. A. Implementation of Merkle and its analyses // Modern scientific researches and innovations. 2017. № 6 URL: http://web.snauka.ru/issues/2017/06/83971

2. Iavich M., Gnatyuk S., Arakelian A., Iashvili G., Polishchuk Y., Prysiazhnyy D. (2021) Improved Post-quantum Merkle Algorithm Based on Threads. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education III. Advances in Intelligent Systems and Computing, vol. 1247. Springer, Cham. https://doi.org/10.1007/978-3-030-55506-1_41. (WoS, Scopus)

3. Improvement of Merkle Signature Scheme by Means of Optical Quantum Random Number Generators; M. Iavich, A. Gagnidze, G. Iashvili, T. Okhrimenko, A. Arakelian, A. Fesenko; Advances in Computer Science for Engineering and Education III (pp.440-453), 2020; DOI: 10.1007/978-3-030-55506-1_40. (WoS, Scopus)

4. Post-Quantum Digital Signatures with Attenuated Pulse Generator; M. Iavich, R. Bocu, A. Arakelian, G. Iashvili; IVUS-2020; http://ceur-ws.org/Vol-2698/; 2020. (Scopus)

5. Improvement of Implementation of Merkle Crypto System, A. Arakelian, O. Polihenko. Scientific and practical cyber security journal; 2020

The University of Georgia

School of Science and Technology

Arturo Arakelyan

Post-quantum digital signature's asynchronous algorithm

Synopsis

of the thesis submitted in partial fulfillment for the Degree of Doctor of Informatics (PhD) in Cryptography

Tbilisi

2022

The doctoral dissertation has been written at the School of Science and Technology, The University of Georgia.

Academic Supervisor: **Maksim Iavich, Bekar Meladze**

Reviewers: **George Iashvili, Elza Djintcharadze, Lela Mirtskhulava**

The defense of the dissertation will on be held on .
Venue: The University of Georgia, Room #519, Building IV, 77a, Kostava str, 0165, Tbilisi, Georgia.

A copy of the dissertation is available at the library of the University of Georgia.

The synopsis was sent on .

Secretary of the Dissertation Board: Natia Manjikashvili

Introduction

Relevance of the topic. Quantum computers are becoming more and more popular. Consequently, existing systems based on factorization of prime numbers become unsafe and vulnerable. A clear example of such systems is the RSA. Accordingly, arises an issue of creating a secure system or refining existing systems. There are many signature systems, one-time and not only, that will be discussed in this paper.

Recently was published an article in which Google, NASA and the Universities Space Research Association (USRA) signed a partnership with a manufacturer of quantum D-Wave processors.

"D-Wave 2X" - the latest quantum processor containing 2048 physical qubits (quantum discharges, the smallest units of information storage in a quantum computer). 1152 qubits are used in this model of quantum computer to perform calculations. Each additional qubit doubles the search space, thus increasing the computing speed.

Cryptography has a very rich and fascinating history. Its history dates back to 4000 years ago, from the ancient Egyptians, and in the 20th century it played an important role in the course of both world wars.

Cryptography is a science that studies the following methods: confidentiality (inability to read certain information by a stranger), data integrity (inability to change information without a trace), authentication and reliability. Initially, cryptography studied cryptographic methods, or the conversion of plain text into encrypted text using a secret algorithm or "key". Traditional cryptography includes symmetric cryptosystems in which plain text encryption and decryption are performed using the same secret key. Modern cryptography also includes: asymmetric cryptosystems, electronic signature systems, hash functions, key management, hidden information retrieval, and quantum cryptography.

For example, hash functions that are actively used in various signature systems, such as the Merkle's signature system, are also actively used to encrypt various data in a database. For example, the image below shows the password field in the database ("MySQL") is stored, which is hashed using "bcrypt".

The development of processors is advancing, newer, more efficient and multi-thread processors are producing. There are many hash-based algorithms that are resistant to attacks by quantum computers but are ineffective.

The aim of the doctoral dissertation is to present a system that will be resistant to the attacks of quantum computers and at the same time quite effective.

Scientific novelty. The scientific novelty of the dissertation is a new algorithm of the existing post-quantum cryptosystem, with the help of which the system is not only safe, but also much faster and more efficient. The algorithm we present uses CPU threads.

Key Generation: The size must be "H> = 2" to sign "2H" documents with a one "public key". Here a signature and confirmation keys are generated: "Xi, Yi, 0 <= i <= 2H". "X_i" is the signature key, and "Y_i" is the confirmation key.

In order to get the parent node, we need to combine 2 previous nodes (children) and hash them; "A [i, j]" are nodes of the current tree:

" a[1,0]=h(a[0,0] || a[0,1]) ".

Current tree is divided by the number of threads of the CPU. In a loop whose length is equal to the number of threads, we calculate the parent nodes. Suppose we have a "t" thread and a "d" node. Divide the number of "d" nodes by the number of "t" threads: "d / t". "D / t" nodes run in separate threads. Parent nodes are obtained according to (1) image. The obtained "sets" are concatenated and

then their "t" "sets" are divided. This process continues until we get the root of the tree. The root of the tree is the "public key".

Message Signing: To sign a message, we transform it to "n" size by hashing. "H (m) = hash", in order to sign the message, you need to use any one-time "Xany" key, one-time signature, one-time "verification" key and all neighboring nodes.

$$\text{Signature} = (\text{sig} \,||\, \text{any} \,||\, Y_{\text{any}} \,||\, \text{auth}_0, \ldots, \text{auth}_{H-1}).$$

Signature confirmation: In order to confirm the signature, it is necessary to check the signature once with the help of the verification key " a[i, j] ", " auth ", " index ", " any ", " Y_{any} ". If the root of the tree matches the public-key, then the signature is correct.

The aim of the study. The aim of the study is to introduce a post-quantum system that will be resistant to the attacks of quantum computers and at the same time quite effective.

Scientists are actively working to create quantum computers. Quantum computers will be able to factorize large numbers. Consequently, quantum computers can break the "RSA" used by many applications. Hash-based digital signatures are an alternative to RSA. These systems use hash functions. The protection of these systems depends on the collision of hash functions.

Merkel's classical algorithm can be considered as static because it does not depend on the number of processor streams. We offer an algorithm that uses processor streams. Here is the mathematical model of this algorithm and the pseudo code of the algorithm. This algorithm has been tested and its speed results are much better than the classic one.

As the development of these computers is in active phase and with the help of the "Shor" algorithm they can easily hack systems that use number factorization, so many existing systems become vulnerable.

Objects of the study. Research objects are existing classical algorithms as well as hash-based systems.

After the RSA, DSA, and ECDSA hacked by a quantum computer, Internet users are likely to conclude that cryptography is dead; That there is no hope that the data will be inaccessible to

hackers, and they will not be able to falsify this data; Many thoughts that a physical shield was needed to securely store information so that the information would be inaccessible to the hacker. For example, hide USB memory in a well-protected safe. However, once you understand the topic, you will be convinced that it is too early to say that quantum computers can easily break RSA, DSA and ECDSA, or that quantum computers have destroyed cryptography. There are many important classes of cryptography behind RSA, DSA and ECDSA:

1. Hashing-based cryptography. A classic example is Merkel's hash-based public-key system (1979), based on the idea of Lamport and Diffie.
2. Code-based cryptography. A classic example is McEliece's hidden "Goppa" system based on the public-key code (1978).
3. Lattice-based cryptography. An example that has probably aroused the greatest interest is the Hoffstein-Pipher-Silverman NTRU public-key system (1998).
4. Quadratic based cryptography. One interesting example is Patarin's HFE's public-key system (1996), which generalizes Matsumoto and Ima's proposition.
5. Secret-key cryptography. The best example is the Daemen - Rijmen Rijndael cipher (1998), abbreviated AES (Advanced Encryption Standard).

It is believed that the systems described above can withstand classical and quantum computers. So far no one was able to adapt Shor's algorithm, which can break: RSA, DSA and ECDSA. Another quantum algorithm is the Grover algorithm. It is not as fast as the Shor algorithm.

Practical importance. This dissertation is practical because its result is a system that can be used in practice. Was developed a new algorithm based on the existing Merkle signature system, but much more efficient and fast. The results of the new algorithm as well as comparisons with several existing systems are given in the paper.

In the future, it is planned to upload the source code of this algorithm to the remote repository, so that everyone can use this algorithm.

The main results of the research. Our new algorithm can use resources of modern processors, based on threads it makes calculations, so the number of operations and the time of calculation is significantly less. Within the dissertation were solved these tasks:

1. Analysis of current symmetrical and asymmetrical cryptographical algorithms.
2. Comparison of modern classical algorithms with post-quantum algorithms.
3. We've described the process of how the new algorithm, which is based on threads, works.
4. We've compared the performance of the old algorithm with the new one.

The performance improvements of the new algorithm is confirmed by results of experiments. For example, using the processor with 8 threads, the number of operations decreased 2.3 times comparing to the old algorithm. If the processor is with 16 threads, the number of operations decreased 9 times comparing to the old algorithm. Also in this algorithm is improved the time of the ciphering/verifying. From this experiment, we can see that the presented algorithm is 3.57 times faster than the classical algorithm. Consequently, this algorithm is also dynamic. Its speed depends on the speed of the processor and the number of threads.

The structure and volume of the dissertation. The paper consists of: a review, a description of the literature, an introduction, 3 chapters, 37 subsections, a conclusion and a list of used literature. The paper presents 14 graphic images.

Counting operations of Merkel's classical algorithm, fractal algorithm and threads' algorithm. *Average costs of the classical algorithm.* Each node in the tree is ultimately part of the authentication path. So, one of the most effective ways is to calculate the total cost of each node. Suppose we have "2H-h" right (hence left) nodes in a "h" tall tree. If we count each node independently, the cost of all of them will be "2h + 1 - 1" operation. To summarize we will have "2H + 1 = 2N" operation. For each "h ($0 \leq h \leq H$)" height, when summing all the fees, we will have an average of "2H = 2log (N)" mandatory operations.

The result calculation phase consists of an “N” cycle, for each leaf we have s ∈ {0,…, N-1}. During each "s" cycle, the result for the "authentication path" for the "s" leaf is $\boldsymbol{AUTH_i, i = 0, \dots, H - 1}$.

Fractal algorithm time. Since we have an "R - 1" number of trees, the total calculation fee for one cycle will be:

$$T_{max} = 2(R-1) < 2S/s.$$

1. Set l = 0.
2. Output Authentication Path for leaf number l.
3. Next Subtree For each $\boldsymbol{j} \in \{\boldsymbol{1, 2, \dots, S}\}$ for which $EXISTS_j$ is no longer needed, i.e, l = 0 (mod 2^{sj}):
    a. Remove Pebbles in $EXIST_j$.
    b. Rename tree $DESIRE_j$ as tree $EXIST_j$.
    c. Create new, empty tree $DESIRE_j$ (if $l + 2^{sj} < 2^S$).
4. Grow Subtrees For each $\boldsymbol{j} \in \{\boldsymbol{1, 2, \dots, s}\}$: Grow tree $DESIRE_j$ by applying 2 units to the modified treehash algorithm (unless $DESIRE_j$ is completed).
5. Increment l and loop back to step 2 (while $l < 2^H$).

Space of the fractal algorithm. The amount of space required for this algorithm can be determined by the number of existing subfields, desired subfields, and with the help of "tails".

First, we have the existing "R" subfields and the "R - 1 desired" subfields and each of them consists of "2s + 1 - 2" "pebbles". The "desired" subdivision associated with the additional "tail" contains "s · j + 1" "pebbles" at the "j> 1" level. Therefore, we have:

$$SPACE_{max} \leq (2R - 1)\,(2^{s+1} - 2) + R - 2 + s\,(R - 2)\,(R - 1) / 2.$$

In the worst case we will have:

$$SPACE_{max} < 2\,R\,2^{s+1} + S\,R / 2.$$

Average costs of the algorithm based on threads. Now we can calculate the number of sequential operations in the new algorithm and compare it to the classical algorithm. Below are the tables showing the number of operations for the classic and new algorithms, according to the different number of threads.

New algorithm run in the CPU with 8 threads	
Quantity of nodes	Number of consecutive operations
8	3
64	10
1024	130
4096	514
16384	2050
262144	30722
1048576	129026

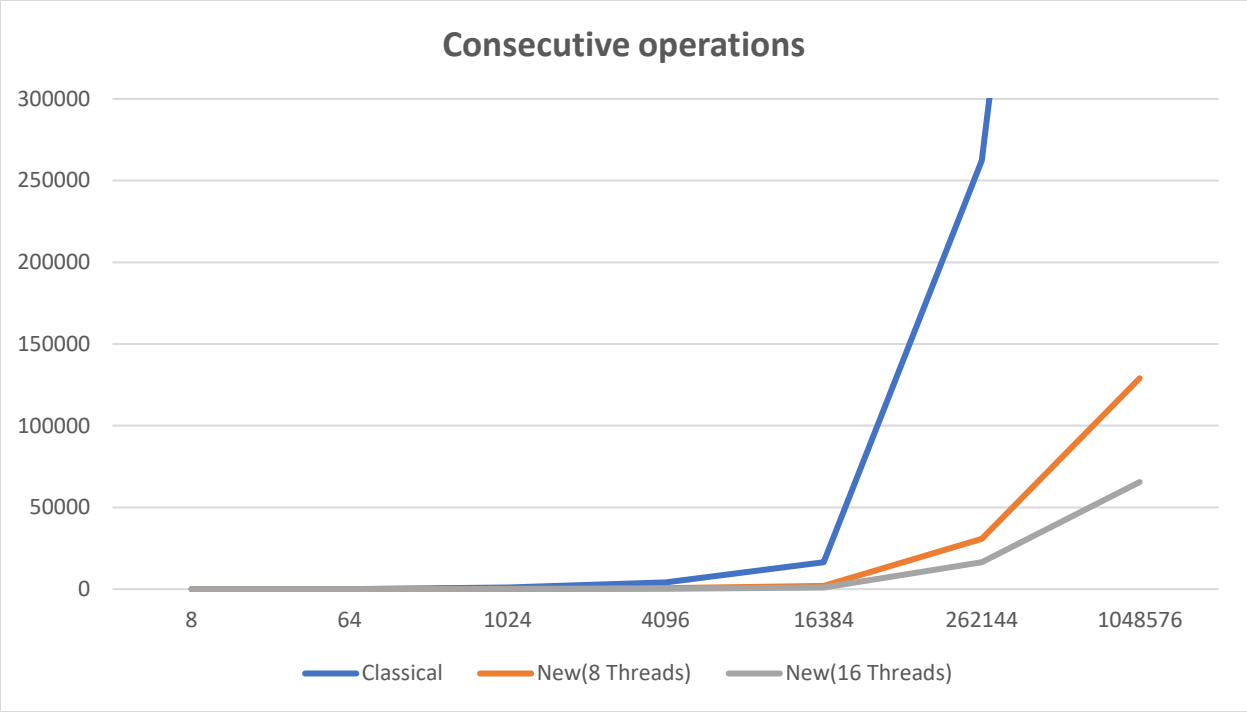
New algorithm run in the CPU with 16 threads	
Quantity of nodes	Number of consecutive operations
8	3
64	7
1024	67
4096	259

16384	1027
262144	16387
1048576	65539

Old algorithm	
Quantity of nodes	Number of consecutive operations
8	7
64	63
1024	1023
4096	4095
16384	16383
262144	262143
1048576	1048575

Therefore, let us derive a formula for calculating the number of sequential operations for a new algorithm. Suppose “t” is the number of threads and "n" is the number of nodes, calculate "O":

$$O = n / t + \log_2 t/2, \text{ თუ } n = t, \text{ then } t = t / 2$$



Conclusion

Analysis. We compared new algorithm based on threads to the classical algorithm. The test was performed on a computer whose processor has 2 threads. The message length was 128 bits.

Results of the classical algorithm:

1. Key generation time: 0.049351 seconds

2. Signature time: 0.0002425 seconds

3. Confirmation time: 0.0038651 seconds

Results of the algorithm based on threads:

1. Key generation time: 0.013841 seconds

2. Signature time: 0.0002425 seconds

3. Confirmation time: 0.0038651 seconds

From this experiment, we can see that the presented algorithm is 3.57 times faster than the classical algorithm.

From the analysis, we can conclude that the algorithm we developed gives us good acceleration during key generation (compared to other existing algorithms). This new algorithm is relevant because today's CPUs are very fast and dynamic. Their development is happening at a very fast pace. Consequently, this algorithm is also dynamic. Its speed depends on the speed of the processor and the number of threads.

List of Publications:

1. Iavich M., Arakeliani. A. Implementation of Merkle and its analyses // Modern scientific researches and innovations. 2017. № 6 URL: http://web.snauka.ru/issues/2017/06/83971

2. Iavich M., Gnatyuk S., Arakelian A., Iashvili G., Polishchuk Y., Prysiazhnyy D. (2021) Improved Post-quantum Merkle Algorithm Based on Threads. In: Hu Z., Petoukhov S., Dychka I., He M. (eds) Advances in Computer Science for Engineering and Education III. Advances in Intelligent Systems and Computing, vol. 1247. Springer, Cham. https://doi.org/10.1007/978-3-030-55506-1_41. (WoS, Scopus)

3. Improvement of Merkle Signature Scheme by Means of Optical Quantum Random Number Generators; M. Iavich, A. Gagnidze, G. Iashvili, T. Okhrimenko, A. Arakelian, A. Fesenko; Advances in Computer Science for Engineering and Education III (pp.440-453), 2020; DOI: 10.1007/978-3-030-55506-1_40. (WoS, Scopus)

4. Post-Quantum Digital Signatures with Attenuated Pulse Generator; M. Iavich, R. Bocu, A. Arakelian, G. Iashvili; IVUS-2020; http://ceur-ws.org/Vol-2698/; 2020. (Scopus)

5. Improvement of Implementation of Merkle Crypto System, A. Arakelian, O. Polihenko. Scientific and practical cyber security journal; 2020