

សៀវភៅបច្ចេក CRYPTOGRAPHY

$$CT = \text{Enc}(M, K)$$

$$M = \text{Dec}(CT, K)$$

$$\text{Sig} = H(M)^d$$

លោកគ្រូ សុខគឹមហេង

២០២០

មេរៀន CRYPTOGRAPHY

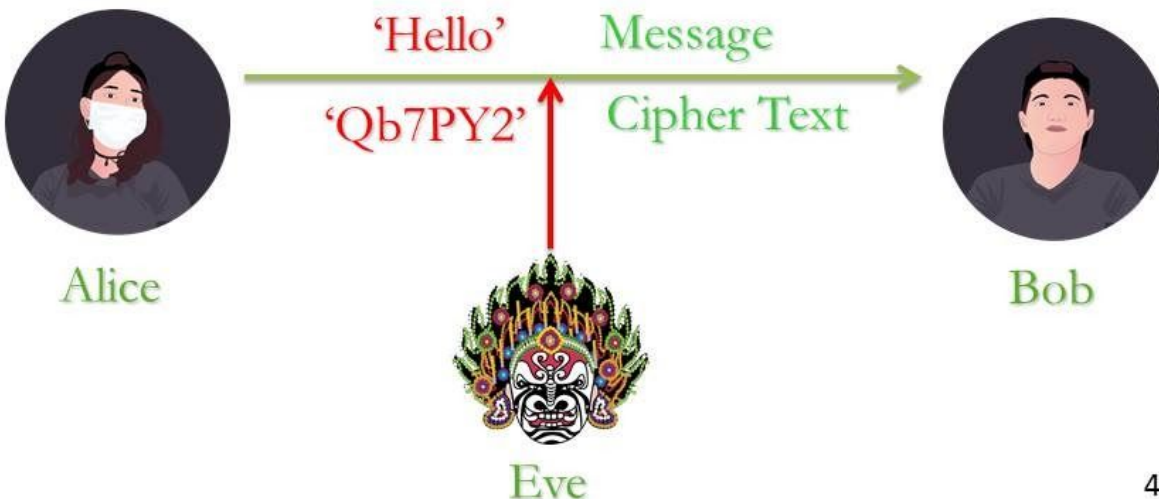
សេចក្តីផ្តើម

Cryptography គឺជាសិល្បៈនៃការបង្កើតការសំងាត់ ដែលមិនចង់អោយអ្នកផ្សេងក្រៅអំពីមនុស្សដែលយើងចង់អោយដឹង។ បើទោះបីជាសម័យបច្ចុប្បន្នសង្គមសព្វថ្ងៃ បានជ្រើតចូលដល់ជីវិតឯកជនរបស់មនុស្សយ៉ាងណាក្តី ក៏បុគ្គល សហគមន៍ ក្រុមហ៊ុន ឬ ប្រទេសមួយ តែងតែមានការសំងាត់ដែលមិនចង់អោយអ្នកដទៃ ក្រៅអំពីរង្វង់នៃភាពទុកចិត្តបានដឹងរៀងសំងាត់របស់ខ្លួន។ Cryptography ត្រូវបានប្រើប្រាស់យ៉ាងទូលំទូលាយក្នុងសម័យសង្គ្រាមលោកលើក ក្នុងការផ្ញើសារសំងាត់ទាក់ទងគ្នា ហើយវាក៏នៅតែមានប្រជាប្រិយភាព ក្នុងសម័យបច្ចុប្បន្នវិទ្យាផងដែរ។ ហេតុដូច្នេះ ដើម្បីអោយកូនខ្មែរ បានដើរទាន់សម័យកាល និង យល់ដឹងពីការបង្កើតការសំងាត់នានា សៀវភៅនេះ នឹងលើកយកបច្ចេកទេស Cryptography មកពន្យល់ជាភាសាជាតិ ដើម្បីអោយងាយស្រួលយល់។

១ និយមន័យ

Cryptography = Crypto (Secret) + Graphy (Write, Study)

Cryptography is the science of writing or creating secret.



Cryptography មកពីពាក្យ Crypto ប្រែថាការសំងាត់ និង Graphy ការសិក្សា ឬ ការបង្កើត សរុបមកគឺជាសិល្បៈនៃការបង្កើតការសំងាត់។ ជាទូទៅ នៅក្នុងមេរៀន Cryptography គេបានបង្កើត តួអង្គមួយចំនួន ដើម្បីងាយស្រួលក្នុងការសិក្សា តួអង្គទាំងនោះឈ្មោះថា Alice ភេទស្រី តួអង្គទី២ ឈ្មោះថា Bob ភេទប្រុស និង តួអង្គជាច្រើនទៀតដូចជា Charlie, Eve, Mallory, Victor, Trent ជាដើម។

ឧទាហរណ៍ Alice ចង់ផ្ញើសារ សូស្តីមួយទៅកាន់ Bob តាមរយៈប្រព័ន្ធអ៊ីនធឺណែត តែដោយសារប្រព័ន្ធ នោះពុំមានសន្តិសុខ ដែលអាចអោយ Eve (Eveasdropper អ្នកលួចស្តាប់) អាចចាប់យកសារដែល Alice បានផ្ញើទៅអោយ Bob។ សារនោះអាចជាឯកសារសំងាត់ លេខកូដធនាគារ ឬ ឯកសារនានាដែល Alice ចង់ប្រគល់អោយ Bob តែម្នាក់គត់ ហើយមិនចង់អោយអ្នកដទៃបានដឹង។ ដោយការមិនជឿទុកចិត្តទៅ លើប្រព័ន្ធអ៊ីនធឺណែត Alice ត្រូវធ្វើអោយសារនោះមើលលែងយល់ ទើបផ្ញើទៅអោយ Bob នៅពេលដែល Bob ទទួលបានសារនោះ ទើបបំប្លែងវាអោយទៅជាសារដើមវិញដើម្បីអានយល់។

ពាក្យបច្ចេកទេស

Message គឺជាសារដែល Alice បានផ្ញើទៅ Bob។ ឧទាហរណ៍ខាងលើ Hello ជា Message។
Cipher Text គឺជាសារដែលបំប្លែងទៅជាអក្ខរដែលមើលលែងយល់។ ឧទាហរណ៍ Qb7PY2 ជាសារ ដែលបានយកពាក្យ Hello ទៅបំប្លែងអោយមើលលែងយល់។
Encryption គឺជាការបំប្លែង Message ទៅជា Cipher Text។
Decryption គឺជាការបំប្លែង Cipher Text ទៅជា Message ដើមវិញ។
Public Key សោរសាធារណៈអ្នកណាក៏អាចមើលបាន។
Private Key ឬ **Secret Key** សោរសំងាត់ មានតែម្ចាស់ព័ត៌មានប៉ុណ្ណោះទើបអាចដឹងពីសារមួយនេះ។

ហេតុអ្វីយើងត្រូវបង្កើតការសំងាត់?

ពីព្រោះយើងមានការសំងាត់ដែលមិនចង់អោយអ្នកដទៃបានដឹង
 ពីព្រោះយើងចង់ចែករំលែកការសំងាត់ទៅមនុស្សដែលយើងជឿទុកចិត្តតែប៉ុណ្ណោះ
 ពីព្រោះយើងមានព័ត៌មានរសើប ដូចជា ការទូទាត់លុយ ព័ត៌មានសុខភាព ការសំងាត់ក្រុមហ៊ុនជាដើម
 ពេលខ្លះការជជែកលេងតាមអ៊ីនធឺណែត ក៏យើងមិនចង់អោយអ្នកដទៃបានដឹង

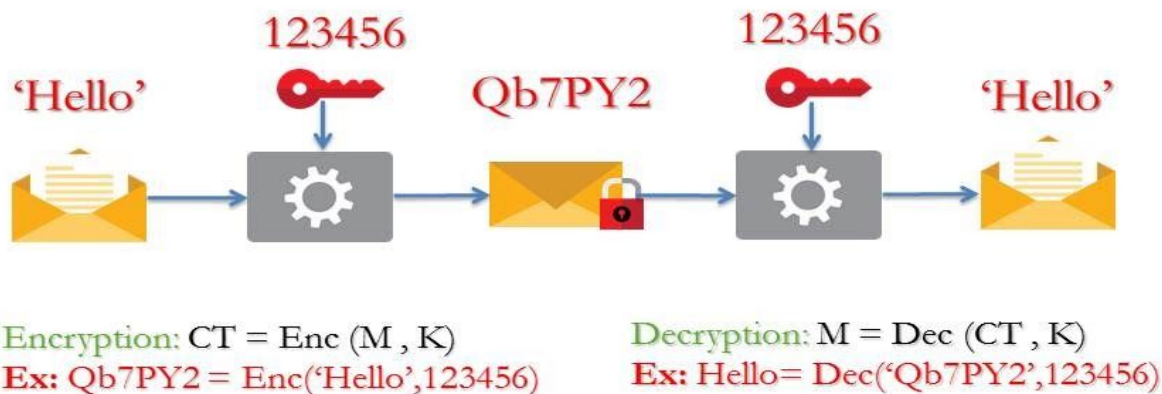
២ ដោះស្រាយរូបមន្តគណិតដ៏លំបាក

នៅក្នុងគណិតវិទ្យា មានបញ្ហាជាច្រើនដែលពិបាកដោះស្រាយ បញ្ហាដែលពិបាកទាំងអស់នោះ ជាធាតុយ៉ាងសំខាន់ក្នុងការបង្កើតការសំងាត់។

- បញ្ហា Discrete logarithm គេអោយតម្លៃ g, g^x គឺពិបាកក្នុងការរកតម្លៃ x ស្មើប៉ុន្មាន។
- បញ្ហា Computational Diffie Hellman (CDH) គេអោយតម្លៃ g, g^x, g^y គឺពិបាកក្នុងការរកតម្លៃ g^{xy} ស្មើប៉ុន្មាន។
- បញ្ហា Decisional Diffie Hellman (DDH) គេអោយតម្លៃ g, g^x, g^y, g^z គឺ $z = xy$ ឬអត់ បើស្មើបញ្ជូនតម្លៃ 0 ឬមិនស្មើទេបញ្ជូនតម្លៃ 1។

III Symmetric Cryptography

Symmetric Cryptography អាចហៅបានថា Private Key Cryptography ឬ Secret Key Cryptography ដែលប្រព័ន្ធនេះ ធ្វើការ Encrypt និង Decrypt ឯកសារដោយប្រើសោរតែមួយគឺ Private Key ឬ Secret Key នោះឯង។ ឧទាហរណ៍ ដូចរូបខាងក្រោម សោរ 123456 ត្រូវបានប្រើប្រាស់សម្រាប់ Encrypt និង Decrypt សារ Hello ។



**“Encryption and Decryption
Key is the same”**

10

ការកំណត់ក្នុងរូបមន្ត

$CT = \text{Enc}(M, K)$ មានន័យថា Cipher Text (CT) ជាអនុគមន៍ Encryption នៃ Message (M) និង Key (K)

$M = \text{Dec}(CT, K)$ មានន័យថា Message (M) ជាអនុគមន៍ Decryption នៃ Cipher Text (CT) និង Key (K)

ទំហំសោរ អាចមានប្រវែង 128bits ឬ 256bits

Algorithm: Caesar Cipher, Vigenère Cipher, Enigma, DES, AES, RC4.....

Mode of Operation: ECB, CBC, CFB, CTR, GCM,

ស្វែងយល់បន្ថែមពី Algorithm ទាំងអស់នេះបានក្នុងកម្មវិធី Openssl ។

គុណសម្បត្តិ



- សុវត្ថិភាព មានល្បឿនលឿន

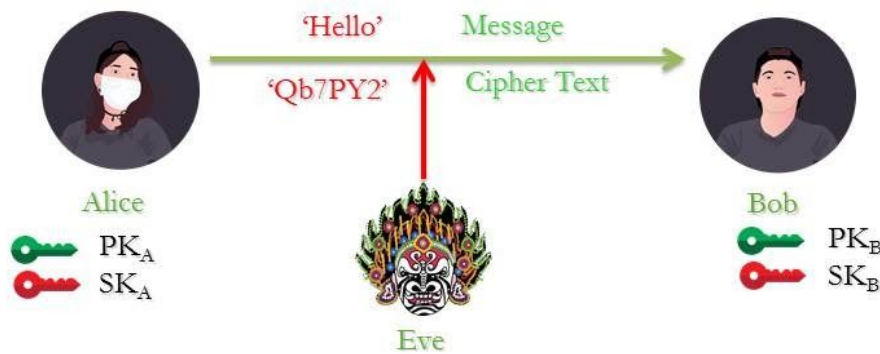
គុណវិបត្តិ

- សោរសំងាត់នៅក្នុងដៃមនុស្សច្រើននាក់
- ត្រូវបញ្ចូលសោរសំងាត់ទៅមក ងាយនឹងជនទី៣លួចបាន

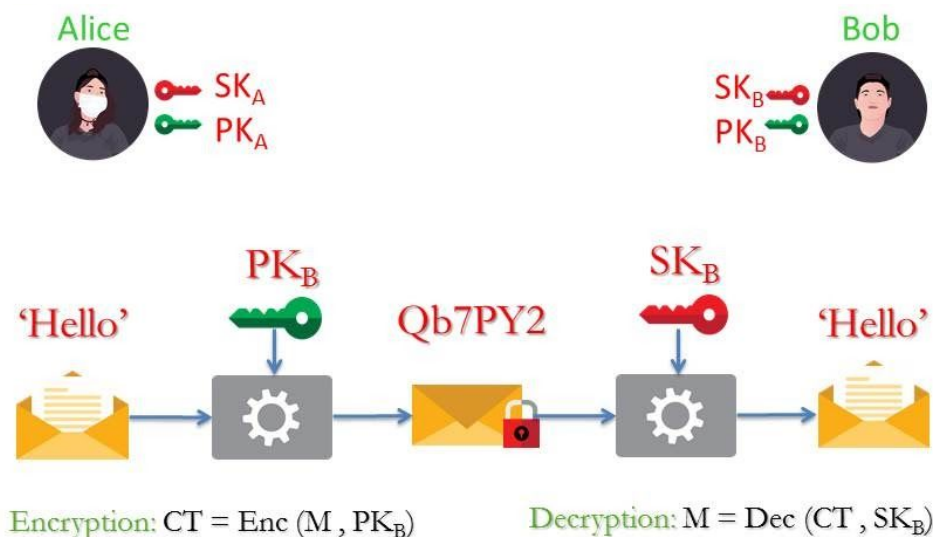
Asymmetric Cryptography

Asymmetric Cryptography អាចហៅបានថា Public Key Cryptography ដែលប្រព័ន្ធនេះ ធ្វើការ Encrypt និង Decrypt ឯកសារដោយប្រើសោរ២ សោរសាធារណៈ Public Key និង សោរសំងាត់ Secret Key។

 Public Key (PK)
 Secret Key (SK) Aka. Private Key



ដូចនៅក្នុងរូបខាងលើ Alice និង Bob មានសោរម្នាក់ៗ។ នៅពេលដែល Alice ចង់ផ្ញើសារទៅអោយ Bob គឺ Alice ត្រូវយកសោរសាធារណៈរបស់ Bob មក Encrypt ជាមួយនឹង Message បង្កើតបានជា Cipher Text។ មានតែ Bob ប៉ុណ្ណោះដែលអាច Decrypt សារសំងាត់ដោយប្រើ សោរសំងាត់របស់ខ្លួន អ្នកដែលគ្មានសោរសំងាត់របស់ Bob គឺមិនអាចបើកមើលឯកសារនោះបានឡើយ។



ការកំណត់ក្នុងរូបមន្ត

$CT = Enc(M, PK_B)$ មានន័យថា Cipher Text (CT) ជាអនុគមន៍ Encryption នៃ Message (M) និង Bob's Public Key (PK_B)

$M = Dec(CT, SK_B)$ មានន័យថា Message (M) ជាអនុគមន៍ Decryption នៃ Cipher Text (CT) និង Bob's Secret Key (SK_B)

ទំហំស្កេរ អាចមានប្រវែង 128bits, 256bits, 512bits, 1024bits, 2048bits, 4096bits

Algorithm: RSA, El Gamal, ECC, Pairing.....

គុណសម្បត្តិ

- សុវត្ថិភាព
- ស្កេរសំងាត់មិនត្រូវបានលួចដោយងាយៗ

គុណវិបត្តិ

- ល្បឿនអាចយឺតជាង Symmetric Cryptography
- ត្រូវការ CPU ខ្ពស់ដើម្បីធ្វើការគណនារូបមន្តនានា

៥ Hash Function

Hash Function គឺជាអនុគមន៍ដែលយកឯកសារដែលមានទំហំប៉ុន្មានក៏បាន ហើយបំប្លែងវាអោយទៅជាអក្សរដែលមានប្រវែងច្បាស់លាស់ ឧទាហរណ៍ SHA256 Function បំប្លែងឯកសារទាំងអស់អោយទៅជា អក្សរដែលមានប្រវែង 256 bits ស្មើនឹង 32 Bytes បើគេសរសេរជាប្រព័ន្ធគោល១៦ Hexadecimal ដែលមានលេខ[0-9][A-F] គឺវានឹងបង្កើតបានជាតួអក្សរ 64ខ្ទង់។

ឧទាហរណ៍

Input	SHA256 Hash Value
123456	8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92
Hello World	a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9f146e
Hello world	64ec88ca00b268e5ba1a35678a1b5316d212f4f366b2477232534a8aeca37f3c

ចំណុចពិសេសរបស់ Hash Function គឺយើងគ្រាន់តែប្តូរមួយតួអក្សរ ប្រៀបដូចជា Hello **W**orld ទៅជា Hello **w**orld យើងនឹងទទួលបាន Hash Value ថ្មីស្រឡាង។

អត្ថប្រយោជន៍

ទី១ គឺគេប្រើប្រាស់ Hash Function ក្នុងការធ្វើអោយ Password មើលលែងយល់ ឧទាហរណ៍ប្រសិនបើ Password យើងមានលេខ 123456 យើងអាចបំលែងវាទៅជា Hash Value 8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92 រួចយកវាទៅរក្សាក្នុង Database នៅពេលដែលគេចូលទៅមើលក្នុង Database គឺគេមិនអាចដឹងថា Password យើងមានលេខ ប៉ុន្មាននោះទេ។

ក៏ប៉ុន្តែពួក Hacker អាចប្រើប្រាស់វិធីសាស្ត្រម្យ៉ាង គឺយក Password ណាដែលគេតែងតែប្រើប្រាស់ ហើយបំលែងវាទៅជា Hash Value បង្កើតបានជា រចនានុក្រមនៃ Password ជាមួយនឹង Hash value រួចធ្វើការប្រៀបធៀប Hash Value ដែលបានបង្កើតនៅក្នុងរចនានុក្រមទៅនឹង Password ដែលមើលឃើញក្នុង Database។ ហេតុដូច្នេះហើយ ទើបទាមទារអោយយើងបង្កើត Password ដែលមានលក្ខណៈ Random ពិបាកទាយ ហើយមិនដែលមានមនុស្សធ្លាប់ប្រើ។ ជាការប្រសើរ យើងអាចយក Private Key ដែលមាន 64 តួអក្សរធ្វើជា Password បាន ពេលនោះ គឺពួក Hacker នឹងពិបាកស្វែងរក ដោយត្រូវចំណាយពេលយូរសែនយូរ។

ទី២ Hash value ក៏អាចជា Fingerprint មួយដើម្បីបញ្ជាក់ថា ឯកសារមួយត្រូវបានកែប្រែ ឬ អត់។ ដើម្បី ងាយស្រួលយល់ខ្ញុំតែងតែលើកយករឿងនិទានខ្មែរមួយ គឺរឿងពុទ្ធសែន(ឫទ្ធសែន) នាងកង្រី យកមកពន្យល់ យក្សសន្ធិមានដែលជាម្តាយរបស់នាងកង្រី បានសរសេរសំបុត្រមួយ អោយឫទ្ធសែន យកមកអោយនាងកង្រី ដោយខ្លឹមសារក្នុងសំបុត្រនោះបានសរសេរប្រាប់នាងកង្រីថា “ទៅដល់ទាំងយប់ សម្លាប់ទាំងយប់ ទៅដល់ទាំងថ្ងៃសម្លាប់ទាំងថ្ងៃ” តែដោយសារឥសីបានដឹងដំណឹង ក៏បានជប់ប្តូរខ្លឹមសារទៅជា “ទៅដល់ទាំងយប់ **រៀបការទាំងយប់ ទៅដល់ទាំងថ្ងៃរៀបការទាំងថ្ងៃ**” ។ នៅក្នុងសម័យបច្ចេកវិទ្យា តើធ្វើយ៉ាងណាទើបដឹងថា ឯកសារមួយមិនត្រូវបានកែប្រែក្នុងបន្តិច ឬ កែប្រែ មុនទៅដល់ដៃអ្នកអាន។ តើធ្វើយ៉ាងណាទើបដឹងថា កម្មវិធីកុំព្យូទ័រមួយជារបស់ម្ចាស់ដើម ហើយមិនពុំ មានមេរោគផ្សេងៗ ដូចជា Spyware ឬ Malware មកជាមួយកម្មវិធីនោះ។ ចម្លើយគឺគេប្រើ Hash Value នេះឯង ឯកសារ ឬ កម្មវិធីដើមត្រូវបាន ដាក់បញ្ចូលទៅក្នុងអនុគមន៍ Hash Function បង្កើតបានជា Hash Value ឬ ហៅថា Fingerprint ដែលសំគាល់ឆ្លុំដើមនៃឯកសារ ប្រសិនបើមានការកែប្រែណាមួយ សូម្បីបន្តិចបន្តួច នោះ Fingerprint នឹងផ្លាស់ប្តូរទាំងស្រុង។ ហេតុដូច្នេះហើយ នៅពេលគេផ្ញើឯកសារ មួយទៅ គេតែងប្រាប់អំពី Fingerprint បន្ថែម ហើយអោយអ្នកទទួលឯកសារ យកឯកសារដែលទទួល បាននោះដាក់ចូលក្នុងអនុគមន៍ Hash Function ដើម្បីទទួលបាន Fingerprint មួយ រួចយកវាទៅ ប្រៀបធៀបជាមួយ Fingerprint ដែលម្ចាស់ដើមបានប្រាប់ ប្រសិនបើលទ្ធផលដូចគ្នា មានន័យថា ឯកសារនោះមិនត្រូវបានកែប្រែទេ ផ្ទុយទៅវិញបើខុសគ្នានោះមានន័យថា នោះមិនមែនជាឯកសារ ដើមឡើយ។

៦ Digital Signature

ហត្ថលេខាឌីជីថល (Digital Signature) គឺជាហត្ថលេខាដែលសំគាល់ម្ចាស់កម្មសិទ្ធិទៅលើឯកសារឌីជីថលណាមួយ។ ប្រសិនបើឯកសារជាក្រដាស គឺយើងយកបិទទៅចុះហត្ថលេខា ហើយពេលខ្លះក៏មានការបន្លំហត្ថលេខា។ ចំពោះហត្ថលេខាឌីជីថល គឺជាការផ្ទៀងផ្ទាត់រូបមន្តគណិតវិទ្យាថាតើម្ចាស់កម្មសិទ្ធិមាន សោរសំងាត់ដែលផ្ទៀងផ្ទាត់ជាមួយនឹង ឯកសារដែលបានចុះហត្ថលេខាដែរឬទេ។

ដើម្បីងាយស្រួលយល់ យើងនឹងលើកយក RSA Digital Signature យកមកពន្យល់៖

យើងមាន p, q ជាចំនួន Prime Number

$$n = p * q$$

Public Key (PK) = e , ដែល $\gcd(e, (p-1)(q-1))=1$

Secret Key (SK) = d , ដែល $d \cdot e = 1 \bmod ((p-1)(q-1))$

Digital Signature = $H(M)^d$ គឺជាតម្លៃ Hash Value នៃ Message (M) ស្វ័យគុណ Secret Key

ដើម្បី Verify ថា Digital Signature ជារបស់ Public Key (e) យើងគ្រាន់តែយក Digital Signature ស្វ័យគុណ Public Key (e) ប្រសិនបើចម្លើយស្មើនឹង $H(M)$ នោះមានន័យថា អ្នកដែលចុះហត្ថលេខាឌីជីថលជាម្ចាស់ Public Key (e) ជាមនុស្សតែមួយ។

ការបង្កើត Digital Signature និង ការផ្ទៀងផ្ទាត់គឺមានច្រើនទម្រង់ទៅតាមរូបមន្តនីមួយៗដែលគេបានបង្កើតឡើង ឧទាហរណ៍ខាងលើ គ្រាន់តែជាប្រភេទមួយនៃ ហត្ថលេខាឌីជីថលមួយតែប៉ុណ្ណោះ។

៧ Diffie Hellman Key Exchange

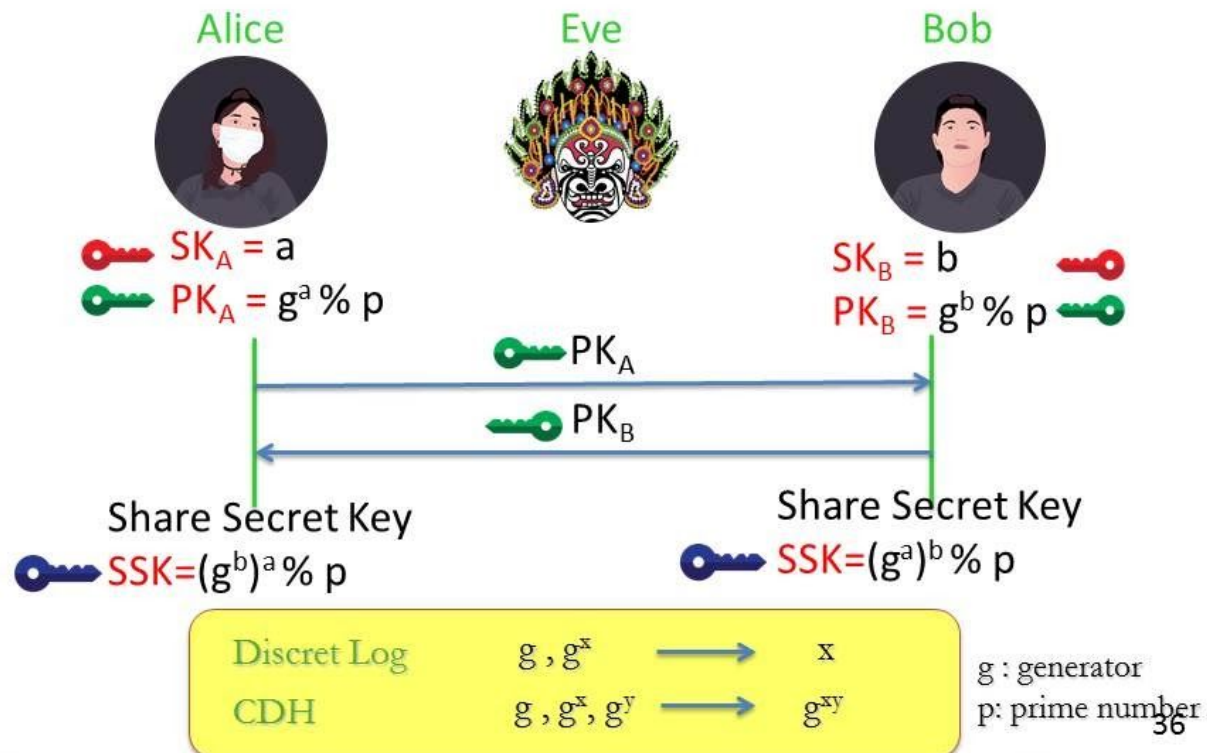
ជាទូទៅ នៅពេលដែលយើងចង់ធ្វើសោរសំងាត់ អោយគ្នាទៅវិញទៅមកតាមរយៈប្រព័ន្ធអ៊ីនធឺណិតដែលគ្មានសុវត្ថិភាព សោរសំងាត់នោះតែងតែត្រូវបានគេលួច។ អ្នកស្រាវជ្រាវពីររូបបានបង្កើតក្បួនច្បាប់មួយ ហើយបានដាក់ឈ្មោះរបស់គាត់ គឺលោក WhiteField Diffie និង លោក Martin Hellman ដែលបានបង្កើត Protocol មួយមានឈ្មោះថា Diffie Hellman Key Exchange។

ឧបមាថា

Alice មាន Secret Key (a) និង Public Key ($g^a \% p$)

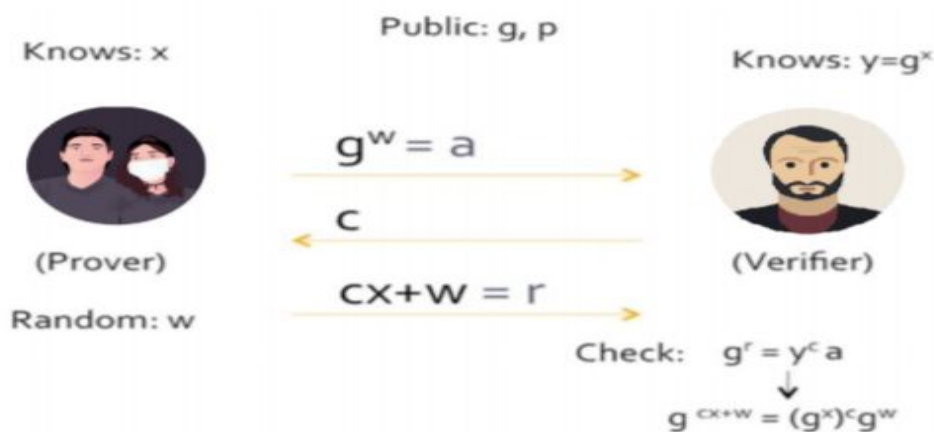
Bob មាន Secret Key (b) និង Public Key ($g^b \% p$)

Alice និង Bob បានធ្វើ Public Key អោយគ្នាទៅវិញទៅមក បើទៅបីជា Eve លួចស្តាប់បានក៏គ្មានបញ្ហាព្រោះវាជា Public Key តែប៉ុណ្ណោះ បន្ទាប់មក Alice និង Bob បានយក Public Key ដែលទទួលបាននោះមកលើស្វ័យគុណជាមួយ Secret Key រៀងៗខ្លួន ដែលទទួលបាន សោរថ្មីមួយហៅថា Share Key ដែលមានទំរង់ជា $(g^{ab} \% p)$ ។ ដោយសារបញ្ហាគណិតវិទ្យាមានការលំបាក ការស្គាល់ g, g^a, g^b គឺមិនអាចគណនា $g^{ab} \% p$ បាន ទើបធ្វើអោយសោរថ្មីរួមនោះ មានតែ Alice និង Bob ទេដែលស្គាល់។



៤ Schnorr Protocol

Schnorr Protocol គឺជាកូនច្បាប់ដែលអនុញ្ញាតិ អោយម្ចាស់កម្មសិទ្ធិ ធ្វើការអះអាងថាខ្លួនជាម្ចាស់នៃកម្មសិទ្ធិទៅលើអ្វីម្យ៉ាង ដោយមិនចាំបាច់បង្ហាញទិន្នន័យទាក់ទងជាមួយនឹងវត្ថុនោះ ដែលគេប្រើប្រាស់ជាហត្ថលេខាឌីជីថលខ្ចាក់ (Blind Signature) ។



ឧបមាថា យើងមានចំនួនសាធារណៈ g និង p
 Prover មានស្រាវស្ទង់ជាតិ Secret Key $= x$
 Verifier មានស្រាវស្ទង់សាធារណៈរបស់ Prover Public Key $= y = g^x$
 ដើម្បី Verifier ដឹងថា Prover ពិតជាម្ចាស់នៃ Public Key (g^x) ដោយមាន Secret Key (x)
 Prover បានបង្កើតចំនួនចៃដន្យមួយ w ហើយបញ្ជូន $a = g^w$ ទៅអោយ Verifier
 Verifier បានផ្ញើ Challenge c មកអោយ Prover
 Prover បានផ្ញើ Response r ដែលស្មើនឹង $cx+w$ ត្រឡប់ទៅអោយ Verifier វិញ
 Verifier ធ្វើការផ្ទៀងផ្ទាត់ តើ $g^r = y^c a^w$ បើផ្ទៀងផ្ទាត់ត្រឹមត្រូវនោះមានន័យថា Prover ពិតជាមានតម្លៃ x
 $g^r = y^c a^w \Rightarrow g^{cx+w} = g^{xc} g^w = g^{cx+w}$

៩ OPENSSL

ផ្នែកនេះយើងនឹងលើកយក កម្មវិធីមួយដែលមានឈ្មោះថា OPENSSL យកមកប្រើដើម្បីធ្វើការងារ ទាក់ទងជាមួយនឹងមេរៀនដែលបានរៀបរាប់ខាងលើ ផ្នែកនេះទាមទារអោយមិត្តអ្នកអាន មាន ចំណេះដឹងផ្នែក Ubuntu Operating System ជាមុនសិន ទើបអាចធ្វើបាន។ ក្រោយពេលដែលយើង បញ្ចូលកម្មវិធី Openssl ទៅក្នុង Ubuntu Operating System រួចហើយ យើងអាចធ្វើតាមលំហាត់ ខាងក្រោម៖

ការបង្កើត Secret Key

```
openssl genrsa -out sk.pem -aes256 4096
```

មើល Secret Key ដែលបានបង្កើតហើយ

```
openssl rsa -in sk.pem -noout -text
```

ការបង្កើត Public Key ចេញពី Secret Key

```
openssl rsa -in sk.pem -out pk.pem -pubout
```

បំលែងឯកសារ file.txt ទៅជា ciphertext.txt

```
openssl rsautl -encrypt -inkey pk.pem -pubin -in file.txt -out ciphertext.txt
```

បំលែងឯកសារ ciphertext.txt ទៅជា file.txt វិញ

```
openssl rsautl -decrypt -inkey sk.pem -in ciphertext.txt > newfile.txt
```

ចុះហត្ថលេខាឌីជីថលលើឯកសារ

```
openssl rsautl -sign -inkey sk.pem -in file.txt -out file_sign.txt
```

ផ្ទៀងផ្ទាត់ឯកសារនិងហត្ថលេខាឌីជីថល

```
openssl rsautl -verify -in file_sign.txt -inkey pk.pem -pubin
```

បង្កើតសញ្ញាប័ត្រឌីជីថល

```
openssl req -new -key sk.pem -out cert.csr -days 365
```

មើលសញ្ញាប័ត្រឌីជីថល

```
openssl req -in cert.csr -noout -text
```

បង្កើត Hash Value

```
openssl dgst -sha256 -out file.hash file.txt
```

ចុះហត្ថលេខាលើ Hash Value របស់ឯកសារ

```
openssl dgst -sha256 -sign sk.pem -out file.hash.sig file.txt
```

ផ្ទៀងផ្ទាត់ហត្ថលេខាឌីជីថលនៅលើ Hash Value របស់ឯកសារ

```
openssl dgst -sha256 -verify pk.pem -signature file.hash.sig file.txt
```

សេចក្តីបញ្ចប់

ដោយសារតែបច្ចេកវិទ្យាសព្ទថ្មីមានការប្រើប្រាស់ Cryptography ច្រើន ក្នុងការបង្កើតការសំងាត់ ការបង្កើតសោរ ហត្ថលេខាឌីជីថលនានា ទើបធ្វើអោយមានការបង្កើតសៀវភៅនេះឡើយ ដើម្បីជាទូន ក្នុងការឈ្វេងយល់ថា អ្វីទៅជា Cryptography តើគេប្រើប្រាស់រូបមន្តគណិតវិទ្យាដូចម្តេច ដើម្បីបង្កើត និង ផ្ទៀងផ្ទាត់។ សព្វថ្ងៃ Cryptography បានដើរទៅមុខយ៉ាងលឿនស្របពេលជាមួយសង្គមបច្ចេកវិទ្យា ដែលត្រូវការប្រព័ន្ធការពារសន្តិសុខ និង ភាពឯកជន។ ហេតុដូច្នេះនេះ ការសិក្សានៅក្នុងសៀវភៅនេះ នឹងអាចបង្ហាញផ្លូវអោយយើងទៅសិក្សាពី មេរៀន Cryptography ដទៃទៀតបានឆាប់រហ័ស។

ជាចុងបញ្ចប់ សូមអភ័យទោសរាល់កំហុសឆ្គងនានាដែលបានកើតឡើងដោយអចេតនា ប្រសិនបើ មានចំណុចកែលម្អ ឬ សំនួរនានាពាក់ព័ន្ធជាមួយនឹងបច្ចេកវិទ្យាមួយនេះ អាចទំនាក់ទំនងតាមរយៈ អាសយដ្ឋានអេឡិចត្រូនិច sok.kimheng@gmail.com សូមអរគុណ។