Заштита информација 2024/2025.

Поштовани студенти,

У оквиру овог пројекта направићете апликацију за кодирање, декодирање и размену различитих датотека одговарајућим алгоритмима, на начин да то буде ваш најбољи рад до сада.

1. Имплементираћете по један алгоритам из сваке групе дате у *Table 1*. Корисник апликације бира којим ће се алгоритмом кодирати фајлови. Механизам за дистрибуцију кључа је произвољан, прихватљиво је све од шапата до Дифи-Хелман шеме.
2. Како бисте аутоматизовали и направили свој рад ефикаснијим, увешћете опцију коришћења *File system watcher*-а (FSW) и повезати фолдер (*Target*) за који ће се детектовати додавање нових фајлова (само додавање, а не и промена и брисање)
3. Ваша апликација треба да подржи и размену кодираних фајлова са апликацијама других студената путем сокета (TCP).
4. На крају, имплементираћете криптографски хеш из *Table 1* који ће служити за верификацију да ли је фајл исправно пренет.

Креирати директоријум *X* у који ће се уписивати криптовани фајлови.

Када је *FSW* укључен прати се додавање нових фајлова у директоријум *Target*. Кад год се дода нови фајл у директоријум *Target*, креирана апликација треба да преузме фајл, кодира садржај одабраним крипто алгоритмом и сними садржај у директоријум *X*.

Када је *FSW* искључен корисник може одабрати било који фајл из било ког фолдера на рачунару за кодирање. Апликација кодира одабрани фајл и кодирани фајл памти у директоријум *X*. Могуће је одабрати опцију декодирања фајлова. Корисник бира неки кодирани фајл, врши се декодирање и декодирани фајл се памти на локацију коју корисник изабере.

Директоријуми *X* и *Target* се дефинишу у поставкама апликације.

Када је укључена опција за размену датотека потребно је повезати се са апликацијом другог студента који је имао исте алгоритме за кодирање. Страна која шаље податак треба одабрати фајл и над њим извршити кодирање. Поред кодирања садржаја фајла, потребно извршити и одговарајући криптографски хеш алгоритам којим ће прималац проверити да ли је фајл исправно пренет. Страна која прима податак треба проверити да ли је фајл исправно пренет и извршити декодирање примљеног фајла. Резултат декодирања треба бити фајл који је послат.

Како би размена фајлова била успешна прво се шаљу подаци о називу и величини фајла (као у примеру датом на рачунским вежбама), а затим дужина резултата примењеног хеш алгоритма у бајтовима и сам резултат хеш алгоритма, а на крају кодирани садржај фајла. Назив фајла се

преноси као стринг, величина фајла је лонг, док се дужина резултата хеш алгоритма у бајтовима преноси као инт, а сам резултат хеш алгоритма као низ бајтова.

Пример за слање података о извршеном хеш алгоритму:

```
byte[] hash = alg.ComputeHash(data);
writer.Write(fileName); // String
writer.Write(fileSize); // Long
writer.Write(hash.Length); // Dužina heš-a
writer.Write(hash); // Niz bajtova
//Upis kodiranog sadržaja fajla
```

Пример за читање података о извршеном хеш алгоритму:

```
string fileName = reader.ReadString();
long fileSize = reader.ReadInt64();
int hashLength = reader.ReadInt32();
byte[] hash = reader.ReadBytes(hashLength);
//Čitanje kodiranog sadržaja fajla
```

Свака апликација треба бити у стању и да шаље и да прима податке.

Студент имплементира алгоритам под редним бројем ***број индекса % 10 + 1***. За имплементацију погрешног алгоритма не добијају се поени.

Table 1: Алгоритми за кодирање

Редни број	Група 1	Група 2	Група 3 - *mod* за групу 2	Група 4 - критографски хеш
1	Railfence cipher	XXTEA	CBC	Tiger hash
2	RC4	XTEA	CBC	BLAKE
3	Playfair cipher	RC6	PCBC	SHA 1
4	Foursquare cipher	LEA	PCBC	SHA 2
5	Double transposition	A5/2	CFB	MD 5
6	Enigma	XXTEA	CFB	Tiger hash
7	A5/1	XTEA	OFB	BLAKE
8	Bifid	RC6	OFB	SHA 1
9	TEA	LEA	CTR	SHA 2
10	Simple substituion	A5/2	CTR	MD 5

Решење: Своје решење предајете као ZIP архиву (искључиво ZIP, пошто остале у великом проценту буду блокиране од стране антивируса и аутоматски уклоњене). Назив архиве мора да садржи број индекса.

Технологија: Можете користити било који програмски језик/окружење за имплементацију, осим програмског језика Python.

Алгоритми: Све алгоритме сами имплементирате.

Дизајн апликације: Када направите дизајн своје апликације, запитајте се „да ли би ово неко користио ако изгледа овако како изгледа“. Код треба да буде написан тако да се избегне дуплирање кода, хардкодирање константи и са минималном цикломатском сложеношћу.

Оцењивање: Оцењиваћемо решење које предајете у предвиђеном року, а одбрана пројекта ће се накнадно организовати.

Референце:

Група 1:

1. Railfence cipher (https://web.archive.org/web/20120105152732/http://cryptogram.org/cdb/aca.info/aca.and.you/chapter_09.pdf#RAILFE)
2. RC4
3. Playfair cipher (https://www.geeksforgeeks.org/playfair-cipher-with-examples/)
4. Foursquare cipher (http://practicalcryptography.com/ciphers/four-square-cipher/)
5. Double transposition
6. Enigma (http://practicalcryptography.com/ciphers/mechanical-era/enigma/)
7. A5/1
8. Bifid (http://practicalcryptography.com/ciphers/classical-era/bifid/)
9. TEA
10. Simple substituion

Група 2:

1. XXTEA (https://en.wikipedia.org/wiki/XXTEA)
2. XTEA (https://en.wikipedia.org/wiki/XTEA)
3. RC6 (https://en.wikipedia.org/wiki/RC6)
4. LEA (https://en.wikipedia.org/wiki/LEA_(cipher))
5. A5/2 (https://medium.com/@shubhamkatheria11/a5-2-ciphering-algorithm-implementation-d594abd06ab8)
6. XXTEA (https://en.wikipedia.org/wiki/XXTEA)
7. XTEA (https://en.wikipedia.org/wiki/XTEA)
8. RC6 (https://en.wikipedia.org/wiki/RC6)
9. LEA (https://en.wikipedia.org/wiki/LEA_(cipher))
10. A5/2 (https://medium.com/@shubhamkatheria11/a5-2-ciphering-algorithm-implementation-d594abd06ab8)

Група 3 – mod за групу 2
(https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation)