getKeyString():

**var stringWriter = new System.IO.StringWriter()** – Létrehozz egy StringWriter példányt, amely lehetővé teszi a szinkron vagy aszinkron írást.

**var xmlSerializer = new System.Xml.Serialization.XmlSerializer(typeof(RSAParameters));** - Létrehozz egy XmlSerializer példányt amelynek paraméterül egy RSAParamétert adunk meg. Paraméterében a typeof operátor egy típustesztelő operátor, amely megvizsgálja az adott osztály megfelel-e paraméternek. Általában egy osztály nevét adjuk át neki, ami által rengeteg információt megtudhatunk az adott osztályról, pl. hogy hány konstruktora van.

Az **RSAParameters**¸ egy osztály, amely egy **ExportParameters** metódust tesz elérhetővé, amely lehetővé teszi a nyers RSA kulcs **RSAParameters** struktúra formájában történő visszakeresést. Ennek a struktúrának a megértéséhez meg kell ismerni az RSA algoritmus működését.

RSA Algoritmus:

Egy RSA kulcspár létrehozásához először keresni kell két nagy prím számot, ez legyen p és q. Ezeket összeszorozzuk és az így kapott eredményt n-nek nevezzük. Mivel p és q egyaránt prímszám ezért n egyetlen tényezője 1, p, q, n,

**xmlSerialize.Serialize(stringwriter, publicKey); -** XML szerializáció, amely egy olyan folyamat ami egy objektum publikus tulajdonságait vagy mezőit konvertálja XML formátummá továbbküldés vagy tárolás céljából.

**Encrypt:**

**var bytesToEncrypt = Encoding.UTF8.GetBytes(textToEncrypt); -** bájt sorozatot állít elő a paraméterben megadott stringből.

Ezt követően létrehozok egy **RSACryptoServiceProvider(2048)**, amely egy aszimmetrikus titkosítást és visszafejtést hajt végre a kriptográfiai szolgáltató által biztosított RSA algoritmus megvalósításával.

**rsa.FromXmlString(publicKeyString.ToString()); -** Ez a metódus kulcsinformációkkal szolgáltatja az RSA objektumot egy XML karakterláncban. Elfogad egy nyilvános kulcsot tartalmazó XML-karakterláncot, vagy egy nyilvános és magánkulcsot tartalmazó XML-karakterláncot. Ezzel a módszerrel jól inicializálhatjuk az RSA kulcsinformációit.

**var encryptedData = rsa.Encrypt(bytesToEncrypt, true);** - az Encrypt metódus hajtja vére magát a titkosítási eljárást arra a byte-okra amiket az első paraméterben adunk át neki. Ha igaz akkor közvetlen titkosítást hajt végre az OAEP padding segítségével.

**var base64Encrypted = Convert.ToBase64String(encryptedData);** - Ez a metódus fogja a titkosított szöveget egy megjeleníthető formára hozni.

**RSAParameters Struct**

Az [RSA](https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.rsa?view=netcore-3.1) algoritmus standard paraméterei.

**RSA algoritmus**

Kulcspár létrehozásához először két nagy prímszámot kell létrehozni, amelyeket p és q neveznek. Ezeket a számokat megszorozzuk, és az eredményt n-nek hívjuk. Mivel p és q egyaránt prímszám, n egyetlen tényezője 1, p, q és n.

[Az RSAParameters](https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.rsaparameters?view=netcore-3.1) semmilyen módon nincs titkosítva, ezért legyünk óvatosak, amikor a privát kulcsokkal kapcsolatos információkat használunk. Valójában a privát kulcsot tartalmazó mezők egyikét sem lehet sorban elrendezni. Ha megpróbál egy [RSAParameters](https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.rsaparameters?view=netcore-3.1) struktúrát távoli hívással vagy valamelyik [sorosító](https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.rsaparameters?view=netcore-3.1) használatával sorba rendezni, akkor csak a nyilvános kulcsokkal kapcsolatos információkat fogjuk megkapni. Ha át akarjuk adni a privát kulcsot, akkor ezeket az adatokat manuálisan kell elküldenünk. Minden esetben, ha valaki le tudja vonni a paramétereket, akkor az átadott kulcs használhatatlanná válik.