



**ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ**

**ФАКУЛТЕТ КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ**

## **КУРСОВА РАБОТА**

**Дисциплина: „Криптографски методи за защита на  
информацията”**

**тема: Криптиране на e-mail**

***Изготвил:***

Кристиян Иванов  
Фак. № 123221009  
Група: 46  
III курс, КСИ, ИТ

***Ръководител:***

ас. Михаела Асенова

**София, 2024**

## Съдържание

1. Въведение в криптографията и необходимостта от защита на информацията...	1
2. Принципи на криптиране на имейли.....	2
2.1. Симетрично криптиране.....	2
2.2. Асиметрично криптиране.....	3
3. Протоколи за криптиране на имейли.....	4
3.1. S/MIME (Secure/Multipurpose Internet Mail Extensions).....	5
3.2. OpenPGP.....	5
4. Имплементация и конфигурация на криптиране на имейли.....	6
5. Сигирност и управление на ключовете.....	7
6. Демо.....	7
7. Заключение.....	12
8. Литература.....	13

## **1. Въведение в криптографията и необходимостта от защита на информацията**

Криптирането на имейл включва криптиране или маскиране на съдържанието на имейл съобщенията, за да се защити потенциално чувствителна информация от четене от всеки, различен от предвидените получатели. Криптирането на имейл често включва удостоверяване.

Имейлът е уязвим носител, особено когато имейлите се изпращат през незащитени или публични Wi-Fi мрежи. Дори имейлите, изпратени в рамките на защитена фирмена мрежа, могат да бъдат прихванати от други потребители, включително идентификационните данни за вход. Криптирането прави съдържанието на вашите имейли нечетливо, докато пътуват от източника до местоназначението, така че дори ако някой прихване вашите съобщения, той не може да интерпретира съдържанието.

Три основни неща, които трябва да криптирате:

- Връзката от вашия имейл доставчик;
- Действителните ви имейл съобщения;
- Вашите съхранени, кеширани или архивирани имейл съобщения.

Криптирането на връзката не позволява на неоторизирани потребители в мрежата да прихванат и уловят вашите идентификационни данни за вход и всички имейл съобщения, които изпращате или получавате, докато напускат сървър на вашия имейл доставчик и пътуват от сървър на сървър в интернет.

Криптирането на имейл съобщенията, преди да бъдат изпратени, означава, че дори ако хакер или някой друг, различен от предвидения получател, прихване вашите имейл съобщения, те са нечетливи и по същество безполезни. Ако съхранявате архивирани имейл съобщения в имейл клиент, като Microsoft Outlook, хакерите могат да получат достъп въпреки защитата с парола на вашите акаунти и дори на вашето устройство. Криптирането на имейлите гарантира, че дори и да бъде получен достъп, съдържанието на вашите имейл съобщения е нечетливо. [1]

## **2. Принципи на криптиране на имейли**

Има два основни типа криптиране на имейли: криптиране със симетричен ключ (симетрично) и криптиране с публичен ключ (асиметрично). Криптирането със симетричен ключ използва един ключ за криптиране и декриптиране на съобщения. От друга страна, криптирането с публичен ключ използва два ключа – частен и публичен – за криптиране и дешифриране на съобщения. [2]

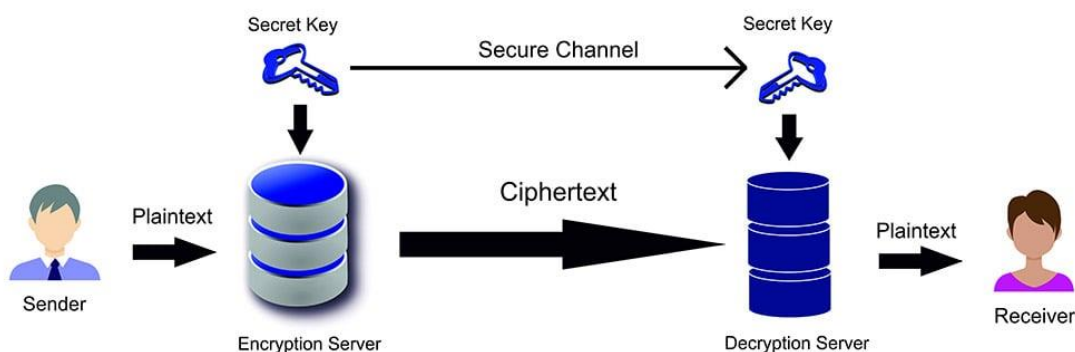
### **2.1. Симетрично криптиране**

Симетричното криптиране е основна техника за защита на данни, където данните се криптират и декриптират с един таен криптографски ключ. Тази методика е използвана още в Римската империя, като исторически пример е шифърът на Цезар. Симетричното криптиране се прилага в различни индустрии, включително отбраната, финансите и здравеопазването, за да защити чувствителна информация. Данните се криптират и декриптират чрез поточни или блокови шифри с един таен ключ. При този процес, подателят криптира данните с ключа, преди да ги изпрати на получателя, който след това използва същия ключ, за да декриптира информацията.

Някои примери за симетрично криптиране включват стандарти като DES, 3DES(Triple DES), AES и IDEA, както и протоколи като TLS/SSL. AES криптирането се издига като един от най-ефективните методи за симетрично криптиране, докато стандартите като DES и Triple DES се считат за устарели и неефективни поради нарастващите заплахи за сигурността. Криптирането на IDEA, въпреки че остава отворен алгоритъм, се счита за остаряло. TLS/SSL протоколът използва симетрично криптиране за защита на данните между клиент и сървър, като генерира уникални сесийни ключове за всяка сесия, които се използват за криптиране и декриптиране на информацията.

Предимствата на симетричното криптиране включват неговата скорост и ефективност. То е бързо и лесно за прилагане, позволявайки криптирането и декриптирането на големи количества данни сравнително бързо. Освен това, алгоритмите за симетрично криптиране като AES се считат за изключително сигурни, като отнемат милиарди години за разбиване с помощта на атаки с груба сила. Така че, за сигурност и бързина, симетричното криптиране е изборът на много организации и индустрии. Недостатъците на симетричното криптиране включват потенциалната опасност от компрометиране на таен ключ при съхранение на него на несигурно място, което може да доведе до разкриване на криптирани данни. Освен това, предаването на този таен ключ между страни или

субекти може да остави комуникацията уязвима на атаки, ако каналът за предаване е компрометиран. Това изисква внимателно управление на ключовете, за да се осигури сигурността на данните. [2]



Фиг. 1. Симетрична криптография [2]

## 2.2. Асиметрично криптиране

За разлика от симетричното криптиране, което използва един и същ таен ключ за криптиране и декриптиране на чувствителна информация, асиметричното криптиране, известно още като криптография с публичен ключ или криптиране с публичен ключ, използва математически свързани двойки публичен и частен ключ за криптиране и декриптиране на подателите и чувствителни данни на получателите. Както при симетричното криптиране, обикновеният текст все още се преобразува в шифрован текст и обратно по време на криптиране и декриптиране, съответно. Основната разлика е, че две уникални двойки ключове се използват за асиметрично криптиране на данни.

В асиметричното криптиране всеки потребител разполага със своя публичен и частен ключ. Когато се изпращат данни, те се криптират с публичния ключ на получателя и могат да бъдат декриптирани само с частния му ключ. Този процес осигурява по-голяма сигурност, тъй като частният ключ не се споделя и остава известен само на получателя.

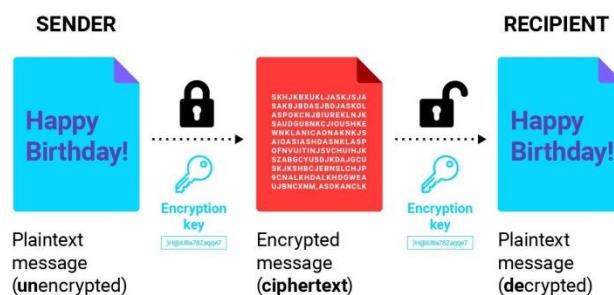
Някои примери за асиметрично криптиране включват RSA, DSA, ECC, метода на Дифи-Хелман и TLS/SSL протокола. RSA, разработено през 1977 г., е един от първите и най-известните примери за асиметрично криптиране, като използва две големи произволни прости числа за генериране на публичен и частен ключ. Друг пример е методът на Дифи-Хелман, който позволява безопасен обмен на ключове през публични канали, като улеснява сигурни комуникации между страни, които не са се срещали предварително.

Сигурността на асиметричното криптиране обикновено се приема за по-висока в сравнение със симетричното криптиране, тъй като не изисква споделяне на таен ключ между страни. Публичните ключове могат да бъдат разпространени безопасно, тъй като декриптирането на информацията, криптирана с публичния ключ, изисква частния ключ, който остава защитен. Това прави асиметричното криптиране по-подходящо за удостоверяване и осигуряване на цялостност на данните. Асиметричното криптиране, използвано в TLS/SSL протокола, позволява установяване на сигурни връзки между клиенти и сървъри в интернет. Публичните и частните ключове се използват за криптиране и декриптиране на данни, като се генерират временни сесийни ключове за всяка връзка, които се използват за основното криптиране на данните.

Предимствата на асиметричното криптиране включват липсата на нужда от разпределение на тайни ключове и възможността за цифрово подписване, докато недостатъците му включват по-бавните операции и възможността за кражба на частни ключове. Разпределението на ключове в асиметричното криптиране е освободено от нуждата от споделяне на тайни ключове, тъй като публичните ключове могат да се разпространяват свободно без риск за сигурността. Също така, асиметричното криптиране позволява цифрово подписване и удостоверяване на автентичността на съобщенията, което е полезно за удостоверяване на авторството и цялостта на данните. Въпреки това, тази технология е по-бавна от симетричното криптиране и изисква внимателно управление на частните ключове, за да се избегнат възможни атаки. [2]

### **3. Протоколи за криптиране на имейли**

Съвременните протоколи за криптиране на имейли играят важна роля в осигуряването на поверителността и защитата на чувствителната информация при обмена на имейли. Тези протоколи предлагат иновативни методи за защита срещу подслушване и неупълномощен достъп до данните, като осигуряват сигурност както по време на предаването на съобщенията, така и при съхранението им на сървърите на доставчиците на имейл услуги. Важно е за лицата и организациите да разбират и да прилагат тези съвременни протоколи за криптиране на имейлите, за да гарантират защитата на своята чувствителна информация и да намалят риска от кибератаки и нарушения на данните. [3]



Фиг. 2. Как работи криптирането на данни [3]

### 3.1. S/MIME (Secure/Multipurpose Internet Mail Extensions)

S/MIME (Secure/Multipurpose Internet Mail Extensions) е стандарт за криптиране и подписване на имейл данни, в съответствие с IETF стандартите. Той използва спецификацията на IETF MIME и стандарта PKCS #7 за защита на съобщенията. S/MIME е вграден в повечето съвременни имейл клиенти и позволява сигурна комуникация между тях, включително разширени цифрови подписи.

S/MIME е широко разпространен протокол за изпращане на цифрово подписани и криптирани съобщения. S/MIME в Exchange Online предоставя следните услуги за имейл съобщения:

- *Криптиране*: Защишава съдържанието на имейл съобщенията.
- *Цифрови подписи*: Потвърждава самоличността на подателя на имейл съобщение. [4]

### 3.2. OpenPGP

OpenPGP е най-широко използваният стандарт за криптиране на имейли. Той е дефиниран от работната група OpenPGP на Работната група за интернет инженерство (IETF) като предложен стандарт в RFC 4880. OpenPGP първоначално е извлечен от софтуера PGP, създаден от Фил Цимерман.

OpenPGP е формат за удостоверяване или криптиране на данни, който използва криптография с публичен ключ и не е патентован. Той е базиран на оригиналния софтуер PGP (Pretty Good Privacy). През началото на 1997 година работната група OpenPGP е създадена в Internet Engineering Task Force (IETF), с цел да дефинира стандарт за този формат, който до тогава беше патентован продукт от 1991 година. През последното десетилетие PGP и

по-късно OpenPGP се превърнаха във стандарт за почти всички подписани или криптирани имейли в света. OpenPGP също така дефинира стандартен формат за сертификати, който, за разлика от множеството други формати на сертификати, позволява мрежи на доверие. [5]

#### 4. Имплементация и конфигурация на криптиране на имейли

Софтуерът за криптиране на имейли е основен инструмент за защита на чувствителна бизнес информация. Най-използваните услуги за криптиране на имейли често са тези, които комбинират сигурността с удобството за потребителите и предлагат надеждност в защитата на данните им. Ето някои от най-популярните и широко използвани услуги за криптиране на имейли:

- **ProtonMail:** Професионална и лична електронна поща, която предлага криптиране на имейли по подразбиране и функции за сигурност като end-to-end encryption.
- **Tutanota:** Услуга за електронна поща, която осигурява end-to-end encryption за всички съобщения и пълна защита на данните на потребителите.
- **Virtru:** Платформа за криптиране на имейли, която осигурява защита на данни в облака и мощни инструменти за контрол и управление на достъпа.

Броят на различните услуги за криптиране на имейли е значителен, обхващайки както големи международни доставчици, така и по-малки специализирани компании или проекти.

Криптирането на имейл клиенти е важен процес, който гарантира сигурността на имейл комуникациите. Това включва конфигуриране на SMTP и IMAP връзки с използване на защитени протоколи като SSL или TLS, което криптира трафика между имейл клиента и сървъра. Така се предотвратява нежеланото четене или прехвърляне на информацията от трети страни.

Генерирането на ключове за криптиране на имейли е важна стъпка в създаването на сигурна комуникационна система. Инструкциите за конфигуриране на различни инструменти и клиенти за имейли включват подробни насоки за генериране на ключове, активиране на криптиране и настройване на други сигурносни параметри. [6]



## 5. Сигурност и управление на ключовете

Управлението на ключове за криптиране на имейли е от решаващо значение за сигурността на данните. То включва създаване, разпространение, съхранение, ротация, отмяна и проверка на ключове, използвани за криптиране и декриптиране на имейл съобщения и прикачени файлове. Без правилно управление, данните могат да бъдат компрометирани, като излагат на риск чувствителна информация и нарушават съответствието с регулациите като GDPR, HIPAA или PCI DSS.

Предизвикателствата пред управлението на ключовете за криптиране на имейли са мащабируемостта, сложността, използваемостта и съответствието с регулаторните изисквания. Разширяването на имейл потребителите, устройствата и домейните може да наложи генерирането и управлението на голям брой ключове. Балансирането на сигурността и удобството на потребителите с различни технически умения и предпочитания също е предизвикателство.

За ефективно управление на ключовете за криптиране на имейли, е важно да се дефинират ясни политики и процедури, да се използва централизирана и сигурна система за управление на ключове, и да се внедри силен процес на управление на жизнения цикъл на криптиращия ключ. Обучението на потребителите и използването на лесни за използване инструменти и поддръжка също са от съществено значение за успешното управление на ключовете за криптиране на имейли. [7]

## 6. Демо

В следващия раздел представям демонстрация на криптиране и декриптиране на имейл с помощта на програмен код на C#. В тази демонстрация ще видите как се криптира и декриптира текстово съобщение, като ще използваме симулация на имейл обмен между два имейл адреса. Кодът използва асиметрично криптиране, като се използва AES алгоритъм.

## EmailEncryption.cs:

```
using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

namespace EmailEncryptionDemo
{
    public class EmailEncryption
    {
        public static byte[] Encrypt(string plainText, byte[] Key, byte[] IV)
        {
            using (Aes aesAlg = Aes.Create())
            {
                aesAlg.Key = Key;
                aesAlg.IV = IV;

                ICryptoTransform encryptor = aesAlg.CreateEncryptor(aesAlg.Key,
aesAlg.IV);

                using (MemoryStream msEncrypt = new MemoryStream())
                {
                    using (CryptoStream csEncrypt = new CryptoStream(msEncrypt,
encryptor, CryptoStreamMode.Write))
                    {
                        using (StreamWriter swEncrypt = new StreamWriter(csEncrypt))
                        { swEncrypt.Write(plainText);}
                        return msEncrypt.ToArray();
                    }
                }
            }
        }

        public static string Decrypt(byte[] cipherText, byte[] Key, byte[] IV)
        {
            using (Aes aesAlg = Aes.Create())
            {
                aesAlg.Key = Key;
                aesAlg.IV = IV;

                ICryptoTransform decryptor = aesAlg.CreateDecryptor(aesAlg.Key,
aesAlg.IV);

                using (MemoryStream msDecrypt = new MemoryStream(cipherText))
                {
                    using (CryptoStream csDecrypt = new CryptoStream(msDecrypt,
decryptor, CryptoStreamMode.Read))
                    {
                        using (StreamReader srDecrypt = new StreamReader(csDecrypt))
                        { return srDecrypt.ReadToEnd();}
                    }
                }
            }
        }
    }
}
```

## Program.cs:

```
using System;
using System.Text;

namespace EmailEncryptionDemo
{
    public class Program
    {
        static void Main(string[] args)
        {
            try
            {
                Console.WriteLine("Please choose an option:");
                Console.WriteLine("1. Encrypt a message");
                Console.WriteLine("2. Decrypt a message");
                int choice = int.Parse(Console.ReadLine());

                if (choice == 1)
                {
                    Console.WriteLine("Please enter the message you want to encrypt:");
                    string original = Console.ReadLine();

                    byte[] key =
Encoding.UTF8.GetBytes("0123456789abcdef0123456789abcdef"); // 32 bytes key
                    byte[] iv = Encoding.UTF8.GetBytes("0123456789abcdef"); // 16 bytes
IV

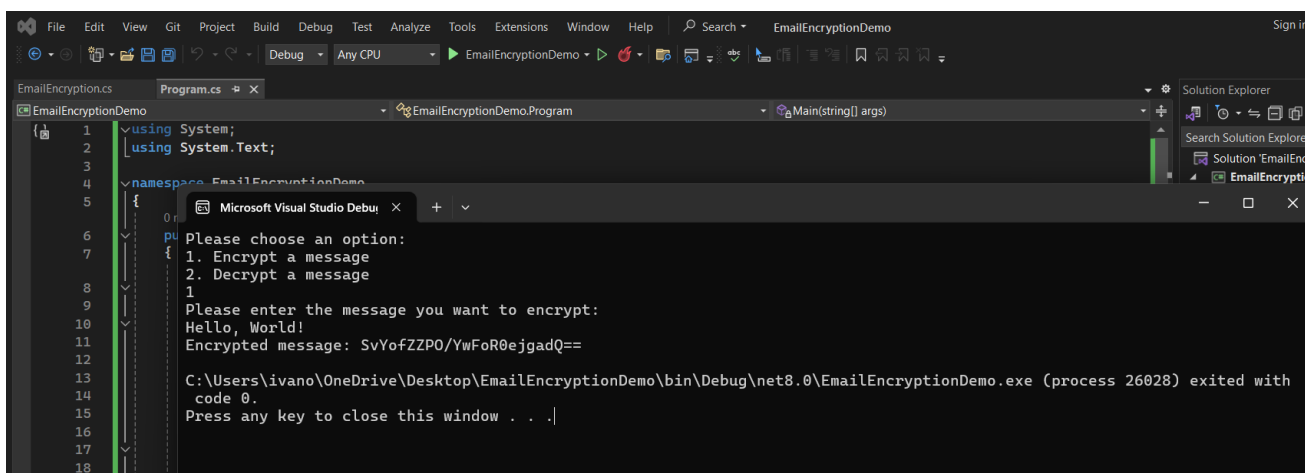
                    byte[] encrypted = EmailEncryption.Encrypt(original, key, iv);
                    Console.WriteLine("Encrypted message: {0}",
Convert.ToBase64String(encrypted));
                }
                else if (choice == 2)
                {
                    Console.WriteLine("Please enter the encrypted message:");
                    string encryptedMessage = Console.ReadLine();

                    byte[] cipherText = Convert.FromBase64String(encryptedMessage);
                    byte[] key =
Encoding.UTF8.GetBytes("0123456789abcdef0123456789abcdef"); // 32 bytes key
                    byte[] iv = Encoding.UTF8.GetBytes("0123456789abcdef"); // 16 bytes
IV

                    string decrypted = EmailEncryption.Decrypt(cipherText, key, iv);
                    Console.WriteLine("Decrypted message: {0}", decrypted);
                }
                else
                {
                    Console.WriteLine("Invalid choice. Please choose 1 or 2.");
                }
            }
            catch (Exception e)
            {
                Console.WriteLine("Error: {0}", e.Message);
            }
        }
    }
}
```

## 1) Криптиране на съобщението

Стартираме приложението, като отворим Visual Studio и изберем проекта "EmailEncryptionDemo". След като приложението се зареди, конзолата ще ни поиска да въведем съобщението, което искаме да криптираме. Въвеждаме съобщението и натискаме клавиш Enter. След като програмата завърши криптирането, ще видим криптирания текст в конзолата.



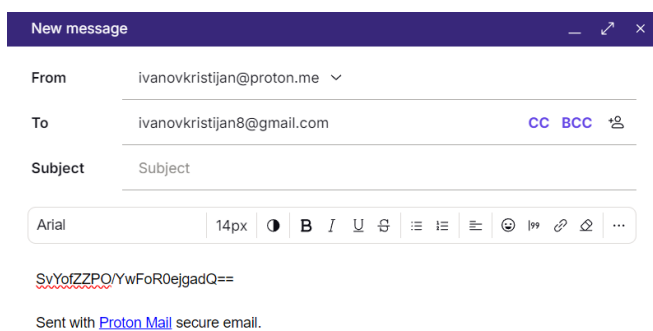
```
using System;
using System.Text;

namespace EmailEncryptionDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Please choose an option:");
            Console.WriteLine("1. Encrypt a message");
            Console.WriteLine("2. Decrypt a message");
            Console.WriteLine("Please enter the message you want to encrypt:");
            Console.WriteLine("Hello, World!");
            Console.WriteLine("Encrypted message: SvYofZZPO/YwFoR0ejgadQ==");
        }
    }
}
```

Скриншот 1.

## 2) Изпращане на криптираното съобщение

След като сме получили криптирания текст в конзолата "SvYofZZPO/YwFoR0ejgadQ==", копираме го. Влизаме в нашия електронен пощенски клиент (например, ProtonMail). Създаваме ново съобщение и в полето за съдържание вписваме копирания криптиран текст. Въвеждаме адреса на получателя и изпращаме съобщението.



New message

From: ivanovkristijan@proton.me

To: ivanovkristijan8@gmail.com CC BCC

Subject: Subject

Arial 14px B I U

SvYofZZPO/YwFoR0ejgadQ==

Sent with Proton Mail secure email.

Send

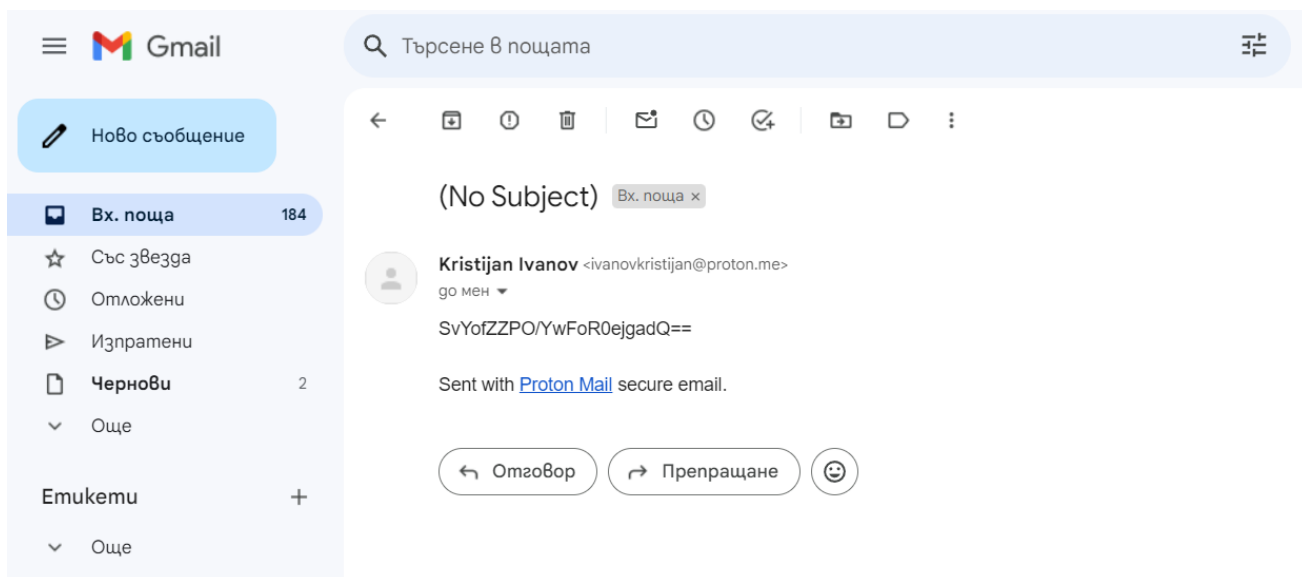
Saved at 11:58 AM

Send

Скриншот 2.

### 3) Приемане на криптираното съобщение от получателя

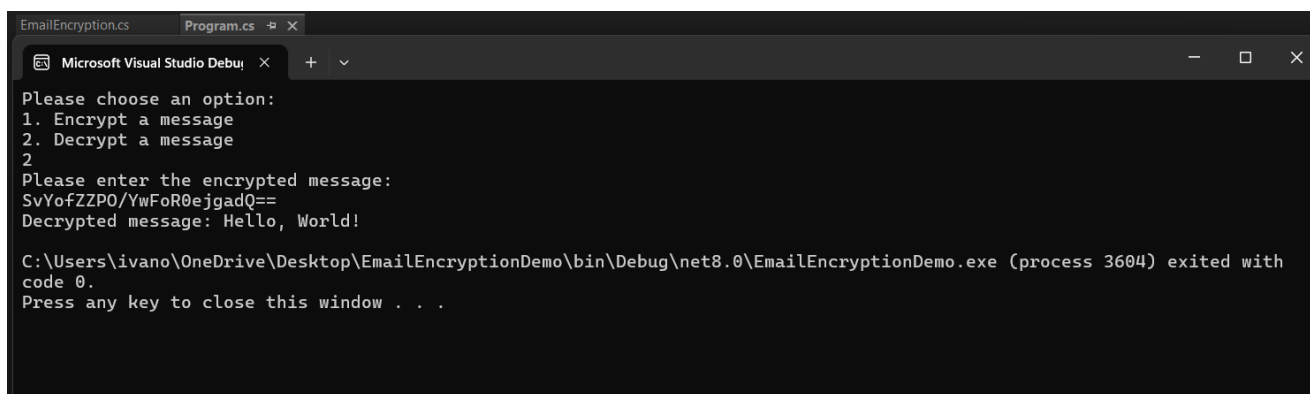
Получателят получава криптираното съобщение в своя пощенски клиент. Отваря съобщението и копира криптирания текст.



Скринишот 3.

### 4) Декриптиране на съобщението от получателя

Получателят използва подходящия софтуер или следва инструкциите, предоставени от изпращача, за декриптиране на криптирания текст. След успешното декриптиране, получателят може да види оригиналното съобщение, което беше изпратено от изпращача.



Скринишот 4.

Линк към GitHub: <https://github.com/kikiivanov/EmailEncryptionDemo>

## 7. Заключение

За бъдещето на криптирането на имейли в света на високотехнологичните заплахи и иновации е от съществено значение организациите да се адаптират и да прилагат най-новите мерки и практики за защита на данните си. Докато имейлите остават важен канал за комуникация между брандовете и техните клиенти, техните уязвимости също така представляват сериозно предизвикателство за сигурността на данните.

С развитието на технологиите за киберсигурност и прилагането на нови стандарти като BIMI (Индикатори на марката за идентификация на съобщения), организациите могат да засилят удостоверяването на автентичността на имейлите си и да предпазват марката си от злоупотреби. Съчетавайки технологии като DMARC и VMC (Сертификати за проверени марки), компаниите могат да укрепят сигурността на имейлите си и да създадат по-доверителни комуникации с клиентите си.

С развитието на технологиите и прилагането на нови стандарти, криптирането на имейли ще продължи да играе ключова роля в сигурността на данните и във възможността за създаване на доверие сред потребителите. [8]

## 8. Литература

- 1) <https://www.digitalguardian.com/blog/what-email-encryption>
- 2) <https://www.trentonsystems.com/en-us/resource-hub/blog/symmetric-vs-asymmetric-encryption>
- 3) <https://www.duocircle.com/email-security/the-evolution-of-email-security-over-time-a-deep-dive-into-eight-modern-email-encryption-protocols>
- 4) <https://learn.microsoft.com/en-us/exchange/security-and-compliance/smime-exo/smime-exo>
- 5) <https://www.openpgp.org/>
- 6) <https://www.pcmag.com/picks/the-best-email-encryption-services>
- 7) <https://www.linkedin.com/advice/0/how-can-you-manage-email-encryption-keys-large-organization-vzgif>
- 8) <https://www.entrust.com/blog/2022/06/the-future-of-email-security-is-here/>