HybridCryptography

تشفير البيانات بطريقة هجينة تم بناءه بالاعتماد على مكتبة crypto الموجودة داخل Node js وبالاعتماد على

- SymmetricCryptography class
- AsymmetricCryptography class

يحتوي تابعين رئيسين تابع تشفير encrypt وتابع فك التشفير decrypt

Example:

```
const hybrid = new HybridCryptography();
const kye = crypto.randomBytes(32);
const iv = crypto.randomBytes(16);
hybrid.setKye(key);
hybrid.setIv(iv);
hybrid.setPublicKey(MyPublickey);
hybrid.setPrivateKey(MyPrivetKey);
hybrid.setReceiverPublicKey(ReceiverPublicKey);
// can encrypt plain text or json object
const data = "...."
const enc = hybrid.encrypt(data)
// default return data as json object if you wont return as text we have add option pramiter false
const dec = hybrid.decrypt(enc)
```

Public Constructors

public `constructor()` source

Public Members

public `asymmetric: Object` source

public `symmetric: Object` source

Public Methods

public `decrypt(data: Object, returnJson: Boolean): Object` source

تابع لفك تشفير الداتا حسب التشفير الهجين
تتم عملية فك التهجين بعدة مراحل

- فك مفتاح التشفير حسب privet key
- فك الرسالة حسب مفتاح التشفير

Params:

Name	Type	Attribute	Description
`data`	`Object`		الابجيكت المراد فك تشفيره
`returnJson`	`Boolean`		متغير افتراضي في حال True يرد الخرج Json والا يرده كنص

Return:

`Object` ; نتيجة فك التشفير

public `encrypt(data: String): object` source

تابع لتشفير الداتا بطريقة هجينة تتم عملية التشفير بعدة مراحل

- توليد key , iv بشكل عشوائي

- تشفير مضمون الرسالة حسب key , iv المولد
- تشفير key , iv حسب publick key للمستقبل

josn وتوابع التشفير المدعومة بالمكتبة لا تسطيع تشفير سوا النصوص لذلك عند ادخل json object نقوم بتحويله الى نص عن طريق عملية stringify
نظامنا يعتمد على تبادل البيانات بشكل object

Params:

Name	Type	Attribute	Description
data	String		النص المراد تشفيره

Return:

object الداتا مشفرة بشكل متناظر مع مفتاح التشفير مشفر بشكل غير متناظر

public getPublicKey(): String source

Return:

String المفتاح العام لي الذي سيتم اعطائه للمستقبل لتشفير مفتاح التشفير المتناظر في الرسالة المرسلة الي

public setIv(iv: Buffer) source

Params:

Name	Type	Attribute	Description
iv	Buffer		مفتاح خاص تتطلبه خورزمية التشفير المستعملة Buffer اذا تمت اضافته بشكل json يتم تحويله بشكل تلقائي الى Buffer يجب ان يكون نوعه

public setKye(key: Buffer) source

Params:

Name	Type	Attribute	Description
key	Buffer		المفتاح الخاص بالتشفير يجب ان يكون نوعه Buffer اذا تمت اضافته بشكل json يتم تحويله بشكل تلقائي الى Buffer

public `setPrivateKey(key: String)` source

Params:

Name	Type	Attribute	Description
key	String		المفتاح الخاص لي الذي سيتم عن طريقه فك مفتاح التشفير المتناظر في الرسالة المرسلة الي

public `setPublicKey(key: String)` source

Params:

Name	Type	Attribute	Description
key	String		المفتاح العام لي الذي سيتم اعطائه للمستقبل لتشفير مفتاح التشفير المتناظر في الرسالة المرسلة الي

public `setReceiverPublicKey(key: String)` source

Params:

Name	Type	Attribute	Description
key	String		المفتاح العام بالمستقبل الذي سيتم تشفير مفتاح تشفير المتناظر بواسطه