genrsa

Golang 生成公私钥,与PHP

补充说明

简介 php对数据进行另密码□golang对数据进行解密

生成公钥私钥对

```
//RSA公钥私钥生成
func GenRsaKey() (pubkey, prvkey []byte, err error) {
    //生成私钥文件
    privateKey, err := rsa.GenerateKey(rand.Reader, 1024)
    if err != nil {
        err = errors.Wrap(err, "rsa.GenerateKey")
        return
    }
    derStream := x509.MarshalPKCS1PrivateKey(privateKey)
    block := &pem.Block{
        Type:  "RSA PRIVATE KEY",
        Bytes: derStream,
    }
    prvkey = pem.EncodeToMemory(block)
    publicKey := &privateKey.PublicKey
    derPkix, err := x509.MarshalPKIXPublicKey(publicKey)
    if err != nil {
        err = errors.Wrap(err, "x509.MarshalPKIXPublicKey")
        return
    }
    block = &pem.Block{
        Type:  "PUBLIC KEY",
        Bytes: derPkix,
    }
    pubkey = pem.EncodeToMemory(block)
    return
}

//公钥加密
func RsaEncrypt(data, keyBytes []byte) ([]byte, error) {
    //解密pem格式的公钥
    block, _ := pem.Decode(keyBytes)
    if block == nil {
        return nil, errors.New("public key error")
    }
    //解析公钥
    pubInterface, err := x509.ParsePKIXPublicKey(block.Bytes)
    if err != nil {
        return nil, errors.Wrap(err, "x509.ParsePKIXPublicKey")
    }
```

```
    //类型断言
    pub := pubInterface.(rsa.PublicKey)
    //加密
    ciphertext, err := rsa.EncryptPKCS1v15(rand.Reader, pub, data)
    if err != nil {
        return nil, errors.Wrap(err, "rsa.VerifyPKCS1v15")
    }
    return ciphertext, nil
}

//私钥解密
func RsaDecrypt(ciphertext, keyBytes []byte) ([]byte, error) {
    //获取私钥
    block, _ := pem.Decode(keyBytes)
    if block == nil {
        return nil, errors.New("private key error")
    }
    //解析PKCS1格式的私钥
    priv, err := x509.ParsePKCS1PrivateKey(block.Bytes)
    if err != nil {
        return nil, errors.Wrap(err, "x509.ParsePKCS1PrivateKey")
    }
    //解密
    data, err := rsa.DecryptPKCS1v15(rand.Reader, priv, ciphertext)
    if err != nil {
        return nil, errors.Wrap(err, "rsa.DecryptPKCS1v15")
    }
    return data, nil
}

//签名
func RsaSignWithSha256(data []byte, keyBytes []byte) (signature []byte, err
error) {
    h := sha256.New()
    h.Write(data)
    hashed := h.Sum(nil)
    block, _ := pem.Decode(keyBytes)
    if block == nil {
        err = errors.Wrap(err, "private key error")
        return
    }
    privateKey, err := x509.ParsePKCS1PrivateKey(block.Bytes)
    if err != nil {
        err = errors.Wrap(err, "ParsePKCS8PrivateKey err")
        return
    }

    signature, err = rsa.SignPKCS1v15(rand.Reader, privateKey, crypto.SHA256,
hashed)
    if err != nil {
        err = errors.Wrap(err, "rsa.SignPKCS1v15")
```

```
        return
    }

    return
}

//验证
func RsaVerySignWithSha256(data, signData, keyBytes []byte) (bool, error) {
    block, _ := pem.Decode(keyBytes)
    if block == nil {
        return false, errors.New("public key error")
    }
    pubKey, err := x509.ParsePKIXPublicKey(block.Bytes)
    if err != nil {
        return false, errors.Wrap(err, "x509.ParsePKIXPublicKey")
    }

    hashed := sha256.Sum256(data)
    err = rsa.VerifyPKCS1v15(pubKey.(rsa.PublicKey), crypto.SHA256,
hashed[:], signData)
    if err != nil {
        return false, errors.Wrap(err, "rsa.VerifyPKCS1v15")
    }
    return true, nil
}
```

测试

使用多语言的例子测试吧，咱们在php这用公钥加密， 在golang用私钥解密：

php:

```
$data = 'hello world!';
$publicKey = '-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDDv2Nvpc2E0Fb9z8SN4S2AyrHb
qSpzHD2uKcEdMR3oBSgrZFaTFQX3WrEU52h+jS4k+Zt6Oo3Lu4poCpcTLNmNQblF
gf/03cq8masVdXaU4AWfjdbbh0ork/ofJhTpt64dT+3Skdl2qSwrntEbyyKFfwue
M/gQKZezE2PvXIz6wwIDAQAB
-----END PUBLIC KEY-----';
$puKey = openssl_pkey_get_public($publicKey);

$encrypted = "";

if (openssl_publicencrypt($data, $encrypted, $puKey)) {
  echo $encrypted;
}
```

```
var token =
`U0hnbOo4lO00Hsuoc4dlN%2FmStkZyMBuI1zYJtVC%2BCdpSVgFFoogQBqHxUhYajOGsyBRZrXF
DFtfRr2vMb3BIIy7wP4WkfbCKTlCk01gf6lxwToFBILuCwy4UNuNrhJ4Elk1jN6wGFen2lgKhNHL
xTuhGgY%2BLqw25rZ85e6uoMqg%3D`
```

```
tk,  := base64.StdEncoding.DecodeString(token)
data, err := util.RsaDecrypt(tk, []byte(setting.Value))
fmt.Println(string(data), err)
```