Introduction to Cryptography, 2021 Fall

Mini report about certificate

0716234 蕭彧

a)

The definition of certificate is an electronic document that used to prove the ownership of a public key. In digital signature, we find a way that can authenticate a message. We hash the message, use sender's private key to encode the hashed message, and concatenate it with the original message. Then, the receiver can use the sender's public key to authenticate the message. The things look perfect, but there is a severe problem. How to know the public key receiver used is truly the public key from the sender? More generally, how to distribute public keys and have a well-defined authentication procedure?

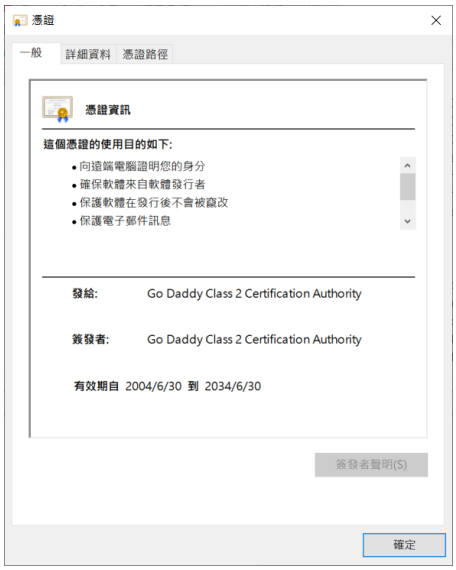
We may first think about use a server to store all public keys in a region. When someone need others' public key, he/she need to request the server and do some handshake to get the keys. It's a good idea if public keys are not so frequently be requested. But, let us consider the https. Whenever one wants to browse a https website, he/she need to request the public key server once to get the corresponding public key. Apparently, it will become the bottleneck of https. We need some offline way to get others public keys. This is the creation of certificate.

We first need to trust some well-known certificate authority (CA) and they will be your root CA. CA will help you authenticate the public key. What it will do is to use its private key to sign a document that contain other's public key, name, uid, to name but a few. At last, it will put its sign into the document. That is, the document here is

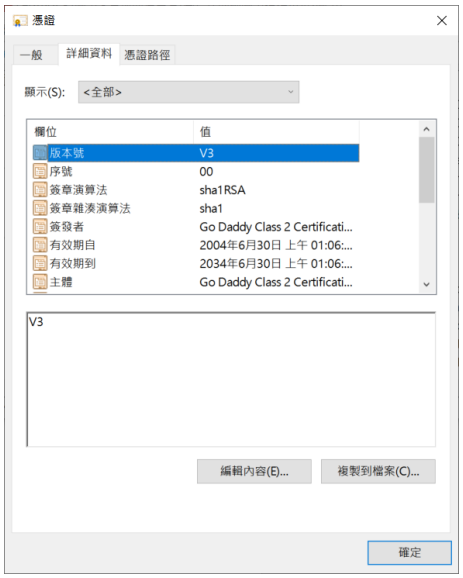
the certificate.

b)

I pick a certificate from go daddy class2 certification authority.



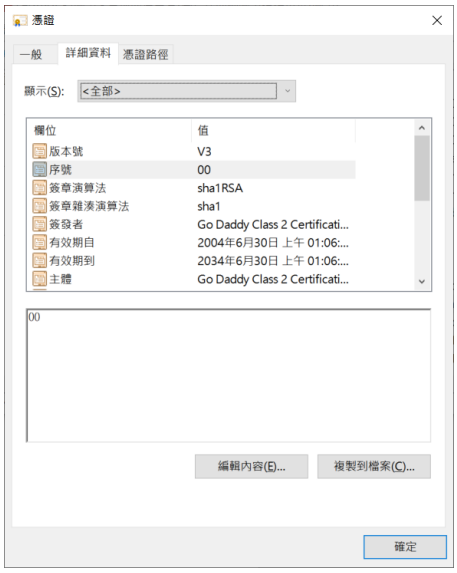
(版本號)



It means it is a X.509 certificate of version3.

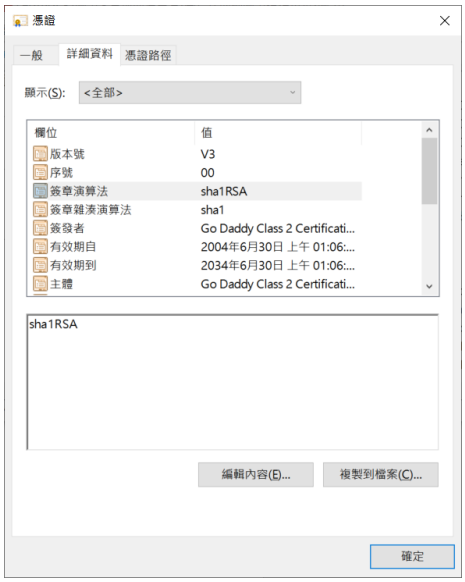
(序號)

It is a unique positive number that CA assign to the certificate.



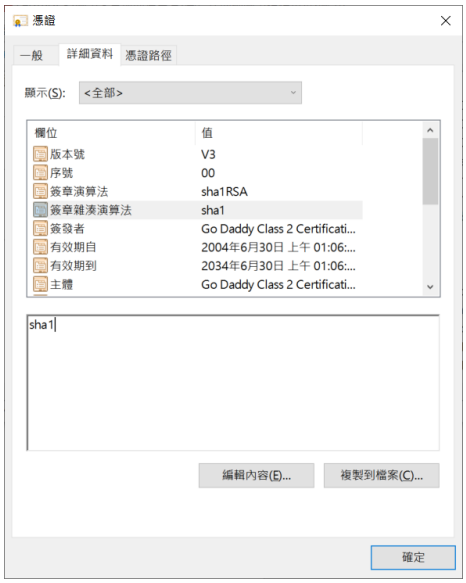
(簽章演算法)

It means the certificate use sha1 for hash and RSA for digital signature.



(簽章雜湊演算法)

It means the certificate use sha1 for hash.



(簽發者)

The name of issuer.

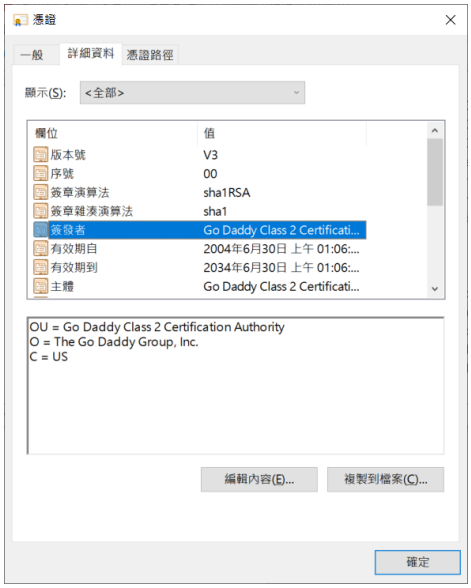
OU = Go Daddy Class 2 Certification Authority

The organization which sign this certificate

O = The Go Daddy Group, Inc.

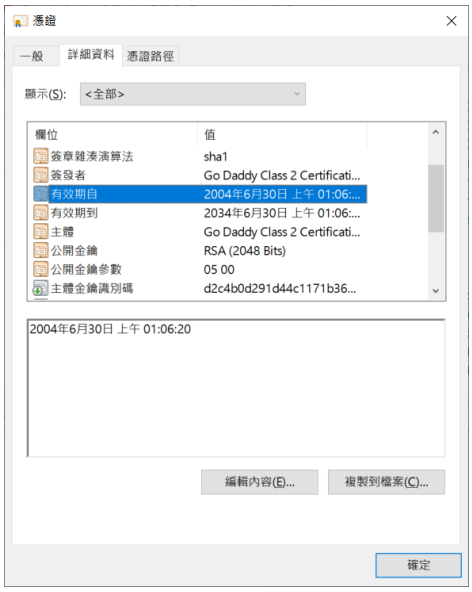
Where the institute which sign this certificate is.

C = US



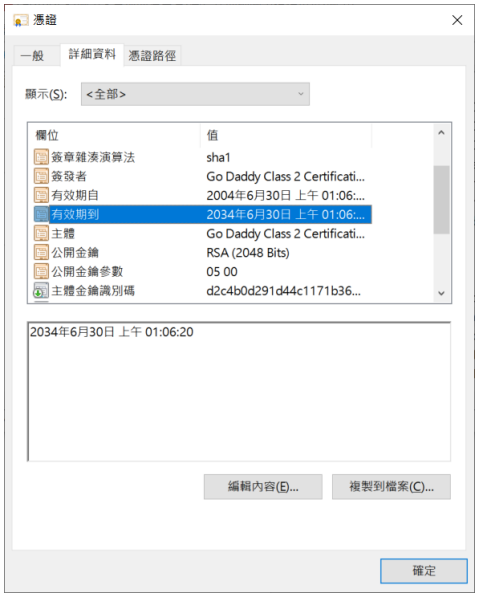
(有效期自)

The date that the certificate start.



(有效期到)

The date that the certificate expired.



(主體)

The name of subject.

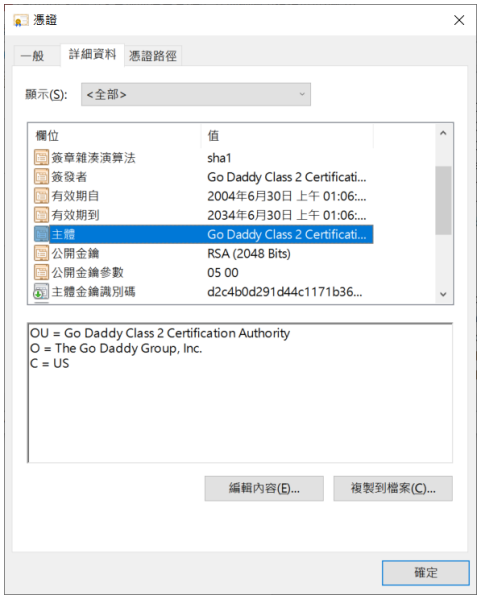
OU = Go Daddy Class 2 Certification Authority

The organization of the subject.

O = The Go Daddy Group, Inc.

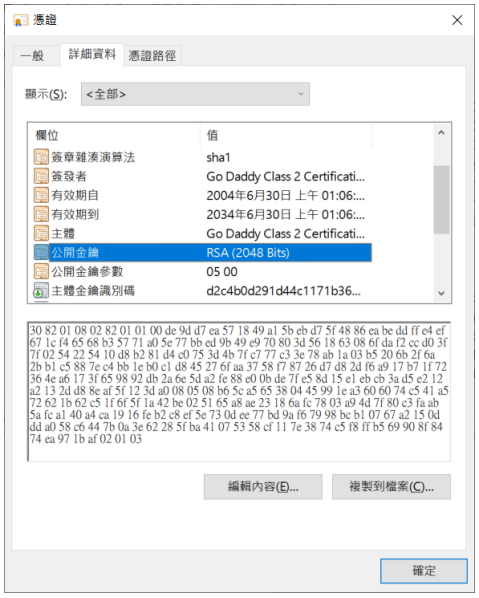
Where the subject is.

C = US



(公開金鑰)

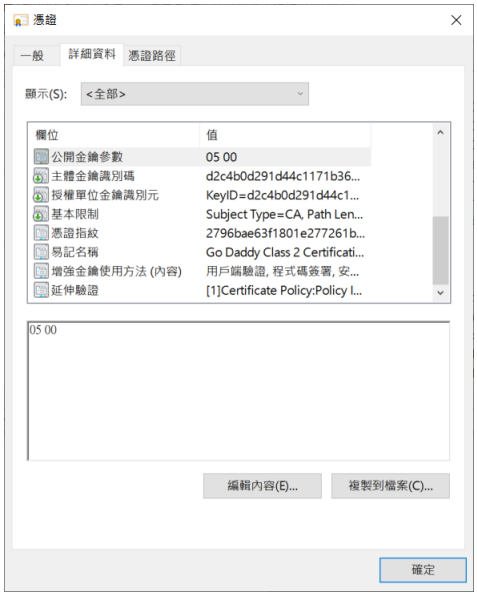
The 2048 bits RSA public key of the subject.



(公開金鑰參數)

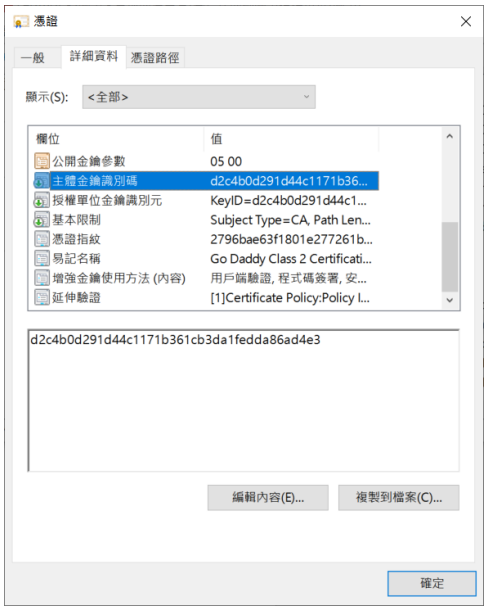
For RSA, the parameters field MUST contain NULL.

Bytes 05 00 mean NULL in DER,CER,BER formats.



(主體金鑰識別碼)

The SKID is used to create the trust chain not based on the certificate subject and issuer but on the certificate SKID and authority key identifier (AKID). This makes it easier to deal with situations where the same subject string is used with multiple CA certificates

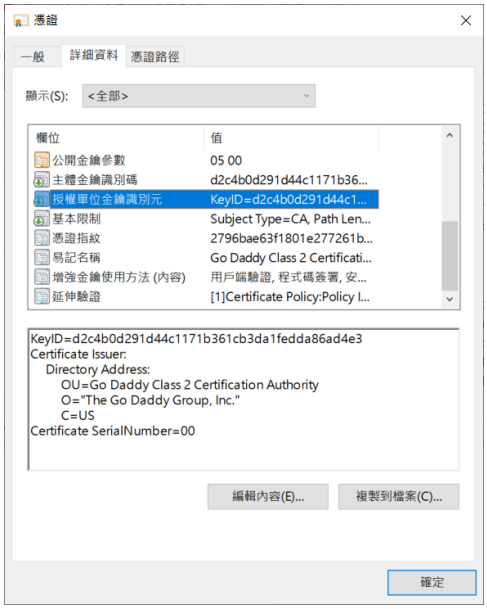


(授權單位金鑰識別元)

AKID extension is used where an issuer has multiple signing keys (either due to multiple concurrent key pairs or due to changeover). The identification may be based on either the key identifier (the subject key identifier in the issuer's certificate) or the

issuer name and serial number.

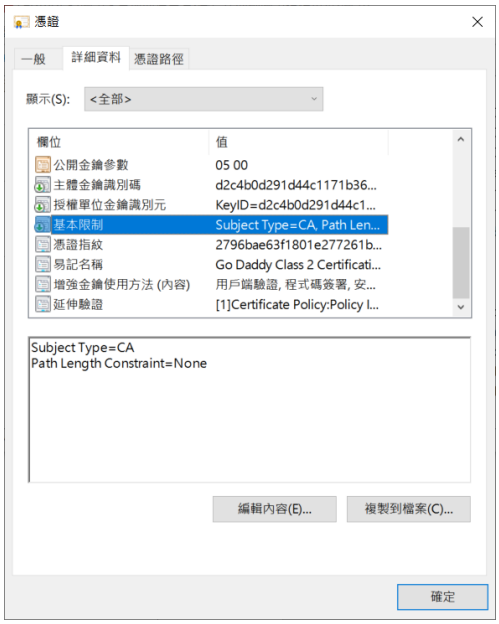
It contains authority key id and some information about authority.



(基本限制)

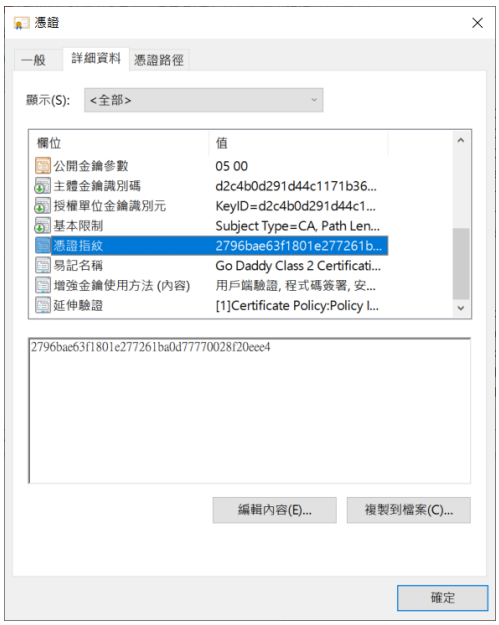
It shows that the subject is CA or not.

For this certificate, it is.



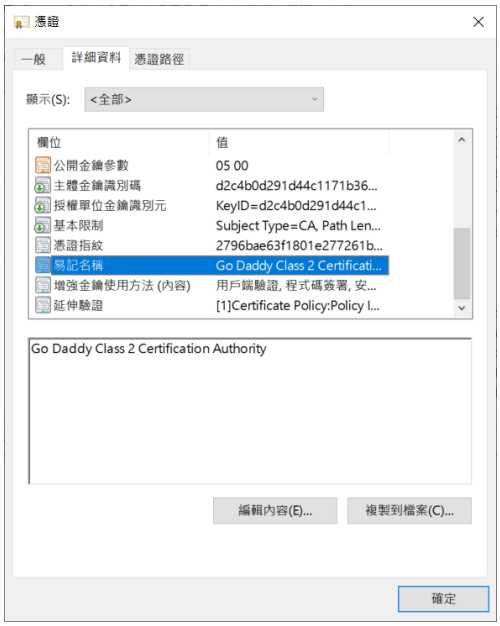
(憑證指紋)

It is the hash value of this certificate. (by sha1)



(易記名稱)

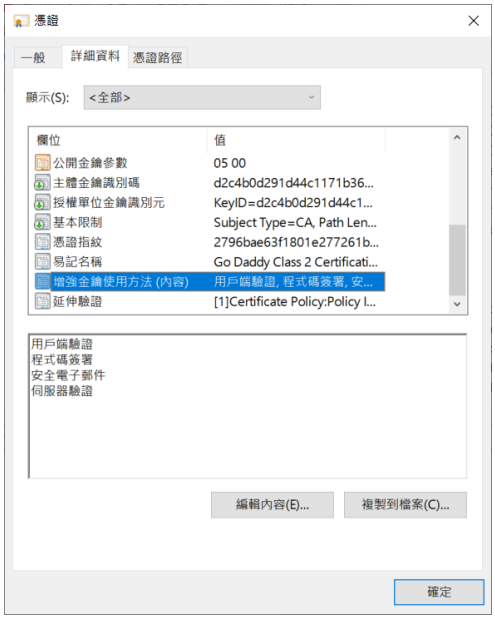
It is alias of the subject.



(增強金鑰使用方法)

The augment usages of the certificate.

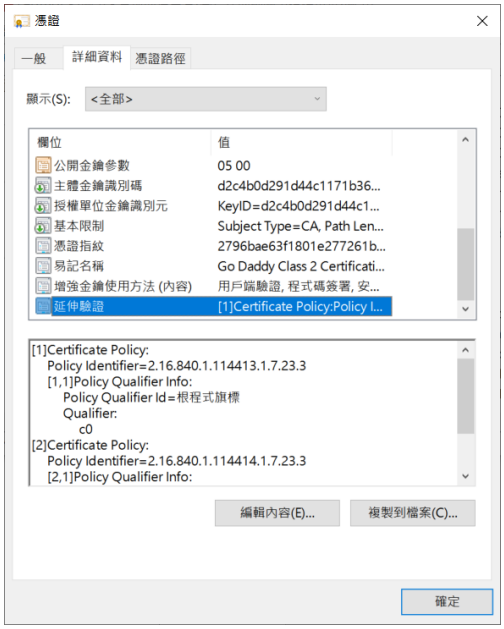
For this certificate, it can do authentication of server, safe email, sign codes and authenticate the users.



(延伸驗證)

Extended validation, the highest security level of the certificate. (EV>OV>DV)

The CA which sign this certificate will strictly check the subject like phoning the subject, require legal documents to prove legal ownership of the registered domain name, to name but a few.



c)

I will talk about the well-known application of certificate, **TLS/SSL,** and an application over it, **HTTPS**, for an example.

TLS stands for transport layer security. It is a security protocol that aim to provide data integrity or confidentiality over Internet. In this protocol, it defines a way that enable client and server build a security channel on Internet. Later I'll talk about the details.

HTTPS stands for HyperText Transfer Protocol Secure. It can be seen as a security version of http. Few decades ago, when people are still using http, every packet on the internet is transparent even the password of somebody. HTTPS is to solve this kind of security problems. When accessing a remote sever with https, one need to do the following things.

1. The browser sends a connection request to the website that you want to connect to, and at the same time asks the website to verify itself.
2. The website sends its own **SSL certificate** back to the client, which contains the public key of the website.
3. The browser verifies the **root certificate** returned by the website, and confirms whether the certificate can be trusted through the chain of trust mechanism, and also confirms whether the certificate has expired.
4. When the authentication is passed, the browser will use the public key of the website to establish a symmetric session key.
5. The website uses its own private key to interpret the session key and sends back a confirmation message to start a session protected by SSL.
6. This session key will be used to encrypt all subsequent data transmitted between

the browser and the website.

The certificate here let client verify the truth of the public key. If client trust the CA of the certificate or the CA of the certificate is in the trust chain of client, client can get the public key, without been attacked by man in the middle attack. Then, the client will use the public key to create a session key. The following communication will be encrypted by this session key, thus enhance security.