西安电子科技大学
XIDIAN UNIVERSITY
计算机科学与技术学院
School of Computer Science and Technology

第二章 数据安全基础

§2.2 公钥基础设施

彭延国

ygpeng@xidian.edu.cn

对称密码、Needham-Schroeder

§2.2.1 Kerberos

§2.2.1 Kerberos - 身份认证的必要性

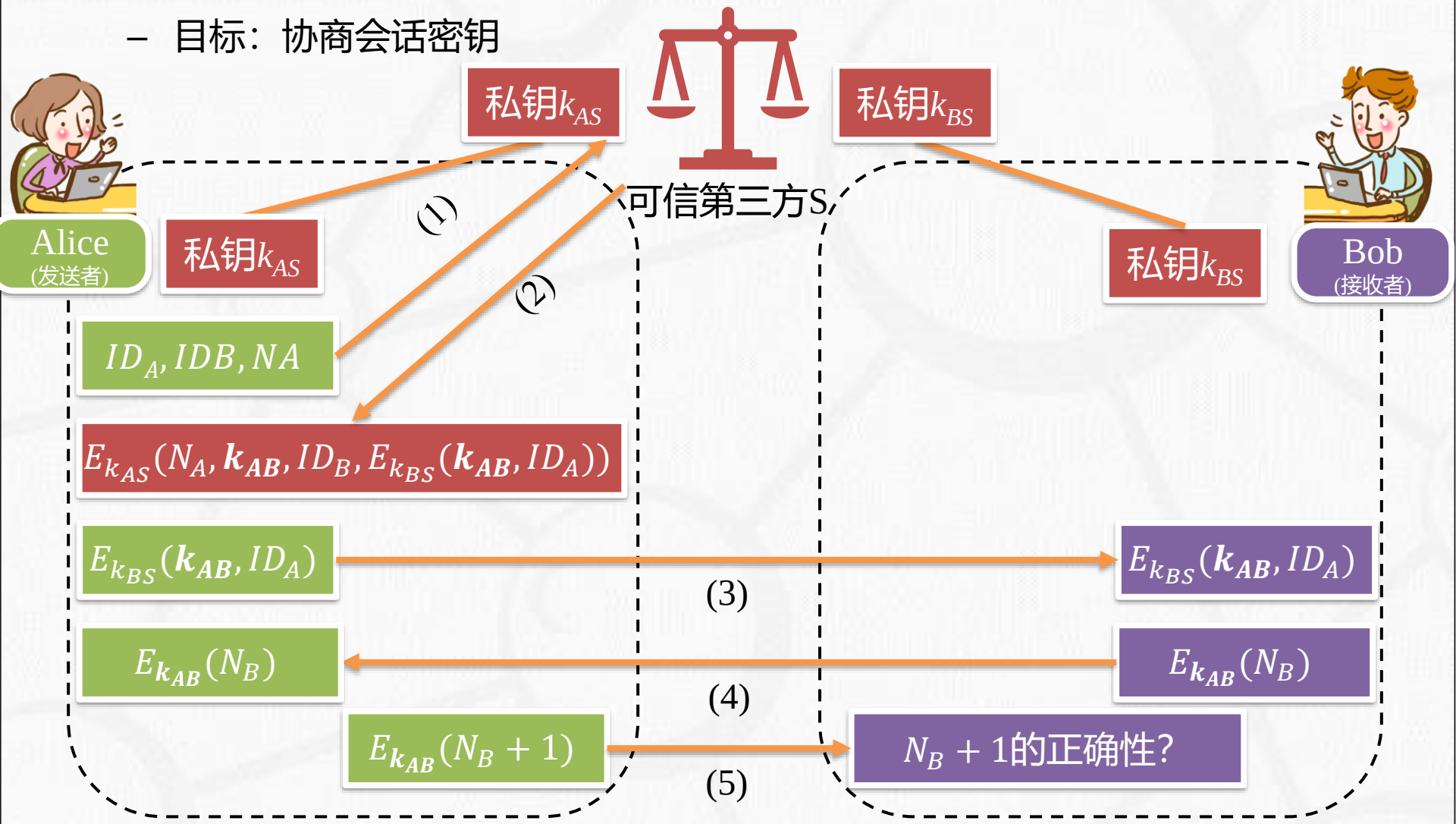
- 身份认证的必要性
  - “我”是“我”；
  - “别人”不是“我”。



- 参与的实体：
  - 服务请求者：向服务提供者申请特定的服务；
  - 身份认证服务器：认证服务请求者的申请，并给合法请求者颁布可用于获取服务的授权令牌；
  - 服务提供者：用于向服务请求者提供特定的服务。
  - 有的时候，身份认证服务器和服务提供者是一个实体。

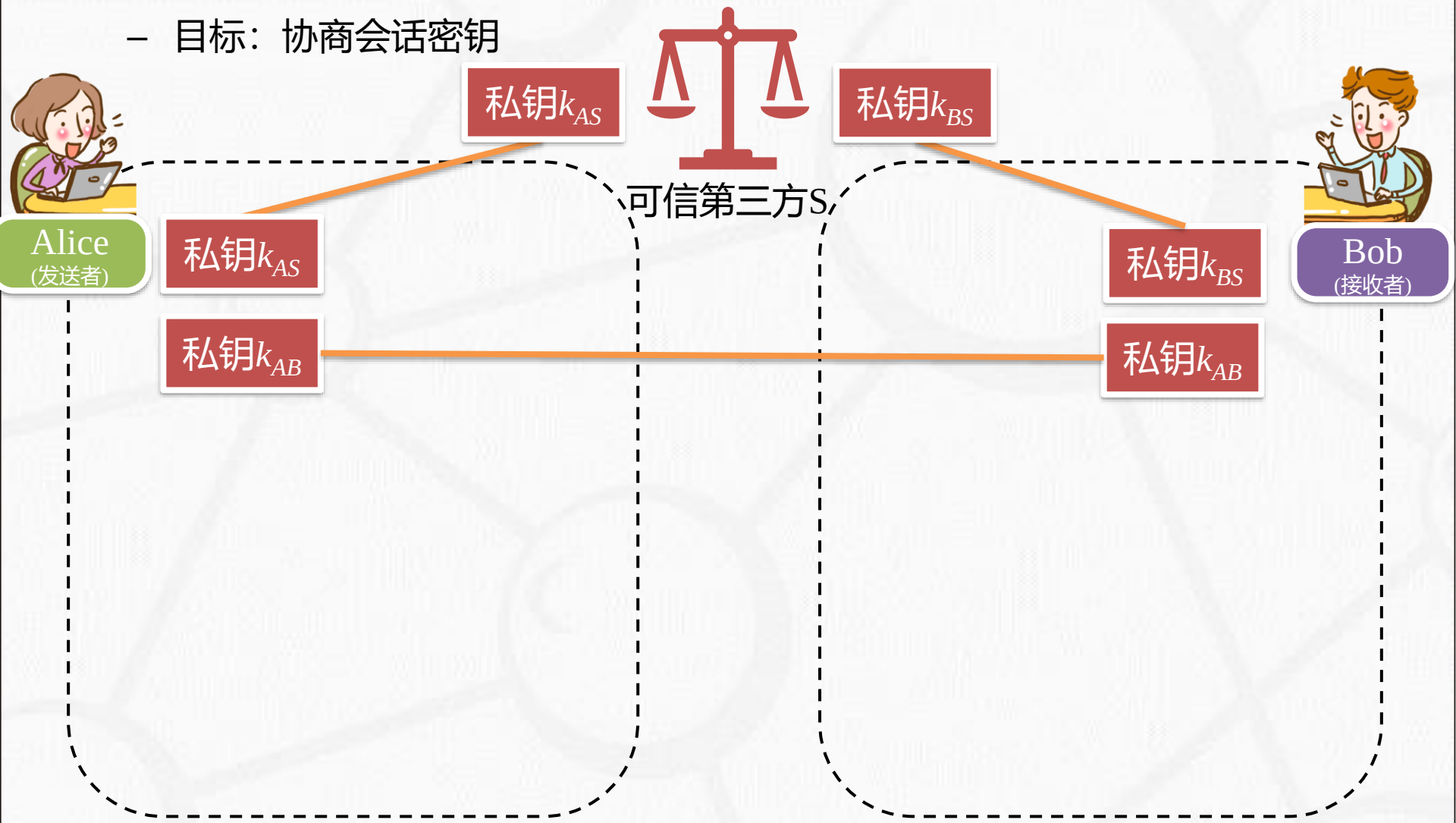
§2.2.1 Kerberos - N-S对称密钥协议

- Needham-Schroeder身份认证协议
  - 目标：协商会话密钥



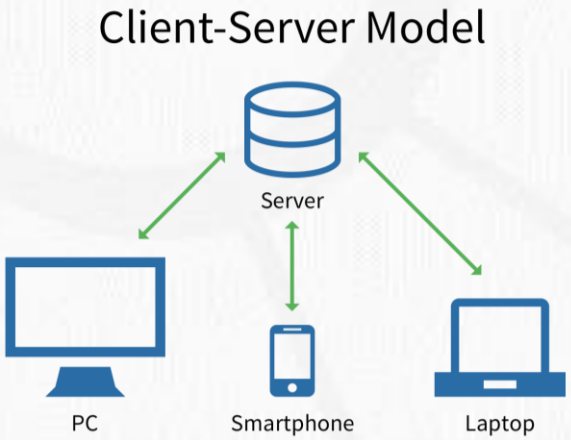
§2.2.1 Kerberos - N-S对称密钥协议

- Needham-Schroeder身份认证协议
  - 目标：协商会话密钥



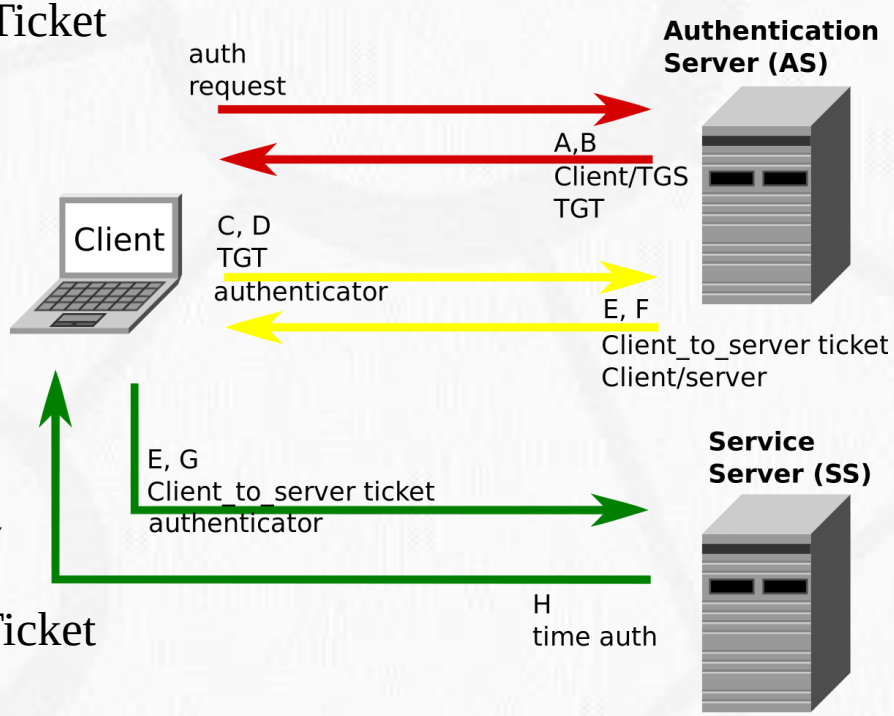
§2.2.1 Kerberos - 简介1

- Kerberos：是一种计算机网络授权协议。
  - 版本v1-v3都只有麻省理工内部发行。v4版本在1988年公开发布。
  - 其设计目标是通过密钥系统为客户机/服务器应用程序提供强大的认证服务。
  - 采用客户/服务器(C/S)模型。
  - 面向局域网环境。
  - 不必信任局域网内所有工作站。
  - 必须都信任认证中心服务器。
  - 允许用户通过访问分布在网络中的服务。
  - 基于Needham-Schroeder的认证协议实现。



§2.2.1 Kerberos - 简介2

- Kerberos参与者
  - 认证服务器(AS: Authentication Service)
    - 维护AS与服务器Server之间的共享密钥
    - 向认证客户C颁发合法的票据Ticket
  - 客户Client
    - 待认证的客户端
    - 服务的需求方
  - 业务服务器(SS: Service Server)
    - 服务的提供方，提供特定服务
    - 验证客户Client发过来的票据Ticket



§2.2.1 Kerberos - 对简单认证会话的攻击

- 一个简单的认证会话：
  - (1) C→AS: $ID_C||P_C||ID_{SS}$
  - (2) AS→C: 票据$Ticket$，$Ticket = E_{K_{SS}}(ID_C||AD_C||ID_{SS})$
  - (3) C→SS: $ID_C||Ticket$
  - 其中：
    - ID_C为客户的身份码，P_C为客户的口令，ID_{SS}为服务器的身份码，AD_C为客户端的网络地址，K_{SS}为业务服务器与认证中心共享的密钥。
- AD_C的作用
  - 防止攻击者A伪装成客户C获取服务。

[1] B Bryant. Designing an Authentication System: A Dialogue in Four Scenes [EB/OL]. MIT. 1988.

§2.2.1 Kerberos - 为什么需要更安全?

- 为什么需要更安全?
  - 口令的明文传输(C→AS: $ID_C||P_C||ID_{SS}$)
  - 针对每个特定服务器，均要一个票据$Ticket = E_{K_{SS}}(ID_C||AD_C||ID_{SS})$
- 如何解决这一问题?
  - 票据授权服务器(TGS, Ticket Granting Service)
    - 提供可重用的票据授权票据(TGT, Ticket Granting Ticket)
    - 用客户的口令生成临时会话密钥
  - AS不再颁发票据Ticket

§2.2.1 Kerberos - 一个更安全的认证会话

- 一个更安全的认证会话

每次客户登录会话就执行一次：

– (1) C→AS: $ID_C||ID_{TGS}$

– (2) AS→C: $E_{K_C}(Ticket_{TGS})$

  - 其中K_C是由P_C生成的，$Ticket_{TGS} = E_{K_{TGS}}(ID_C||AD_C||ID_{TGS}||TS_1||Lifetime_1)$

每种类型的服务执行一次：

– (3) C→TGS: $ID_C||ID_{SS}||Ticket_{TGS}$

– (4) TGS→C: $Ticket_{SS}$

  - 其中，$Ticket_{SS} = E_{K_{SS}}(ID_C||AD_C||ID_{SS}||TS_2||Lifetime_2)$

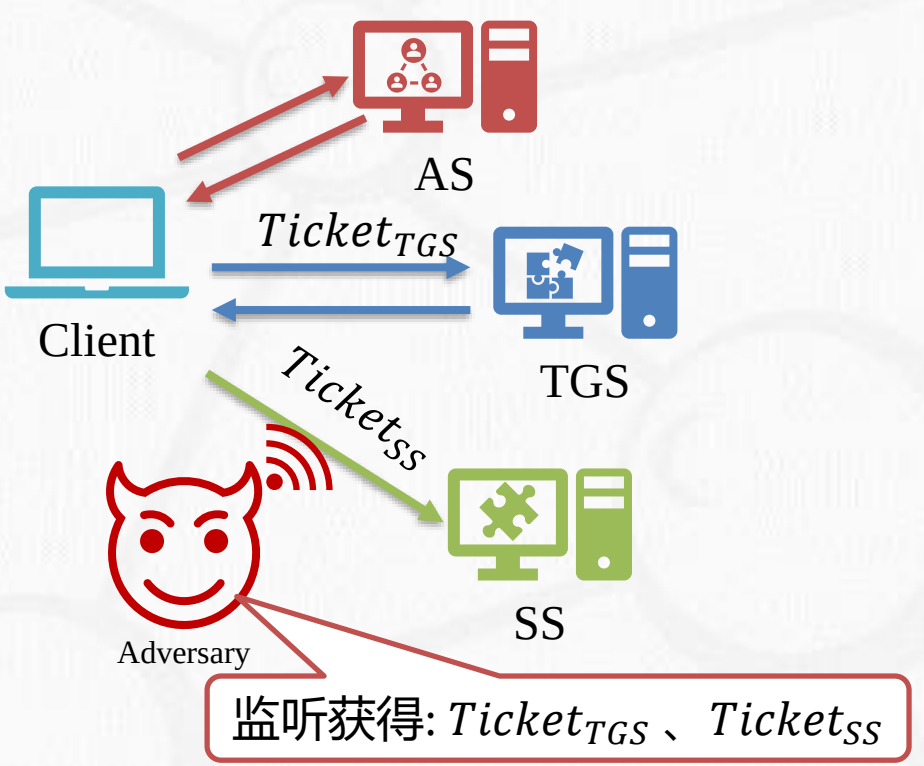
时间戳和有效期

每个服务会话执行一次

– (5) C→SS: $ID_C||Ticket_{SS}$

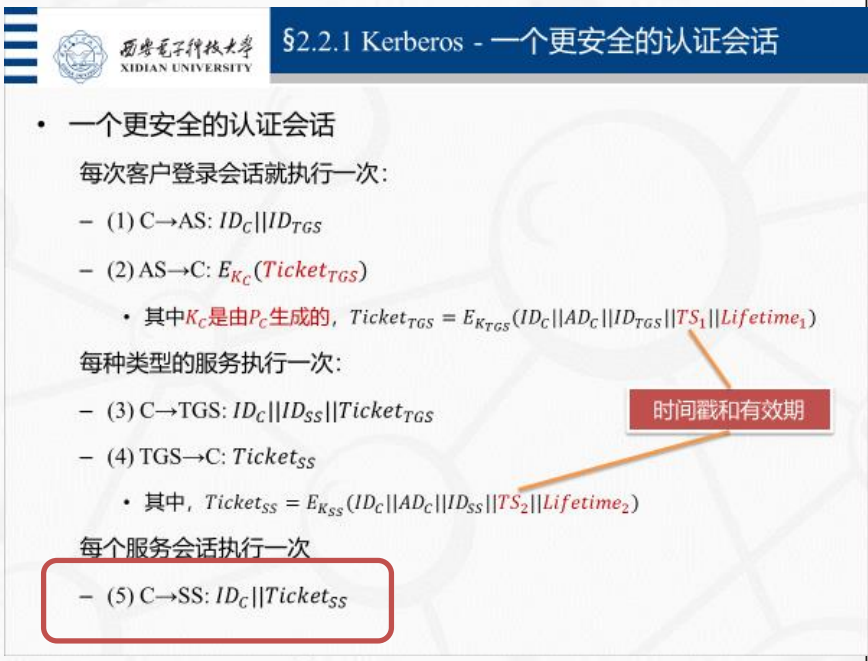
§2.2.1 Kerberos - 票据/TGT的重放攻击

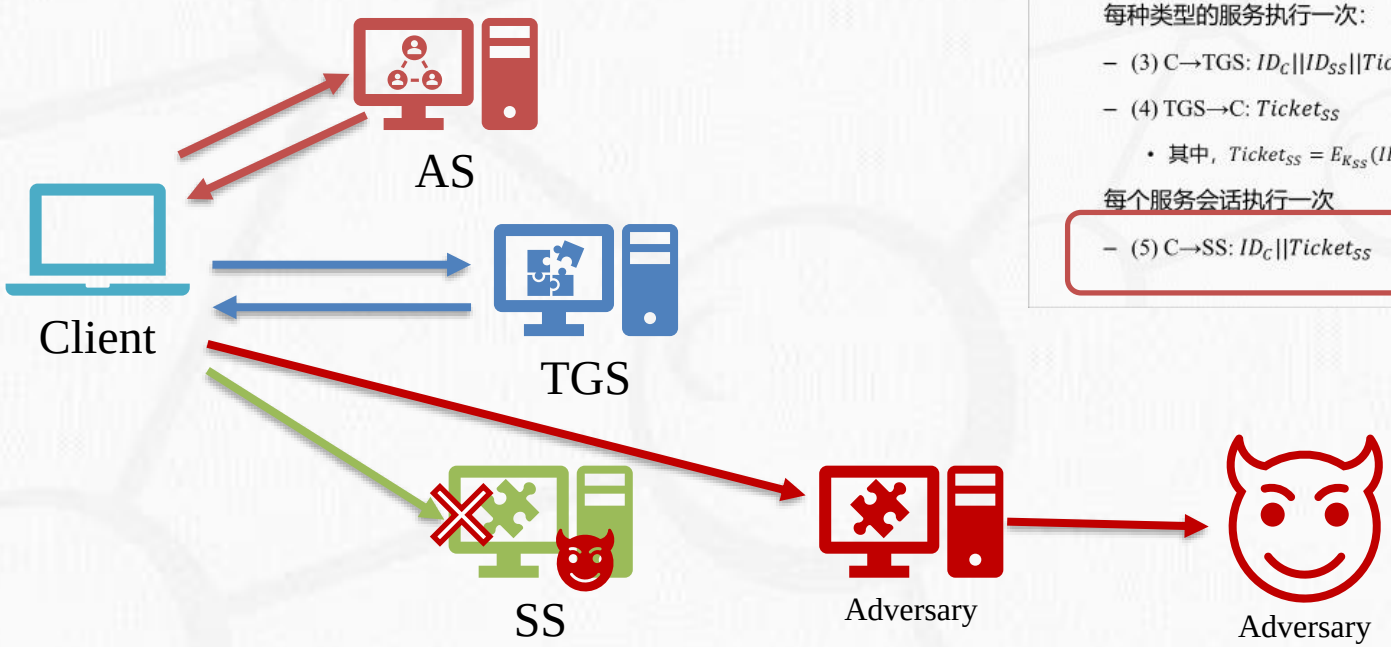
- 这么做足够安全吗?
  - 情形1：票据/TGT的重放攻击
    - 若票据有效期过长，敌手在客户离线后伪装成客户，用相同的网络地址使用截取到的票据/TGT索取未经授权的认证/业务服务。



§2.2.1 Kerberos - 业务服务器的沦陷

- 情形2：业务服务器可能遭遇沦陷攻击
  - 若不对业务服务器的身份进行验证，攻击者可能攻陷业务服务器，并伪装成业务服务器，实施中间人攻击。
    - 获得客户全部信息。





§2.2.1 Kerberos - v4概览

每当用户登录对话时执行一次

1.用户登录工作站，并请求主机服务

请求票据授权票据

票据+会话密钥

请求服务授权票据

票据+会话密钥

每种类型服务执行一次

3.工作站提示用户输入口令，并用口令解密到来的消息。然后将票据和包含用户名、网络地址和时间的认证符发给TGS

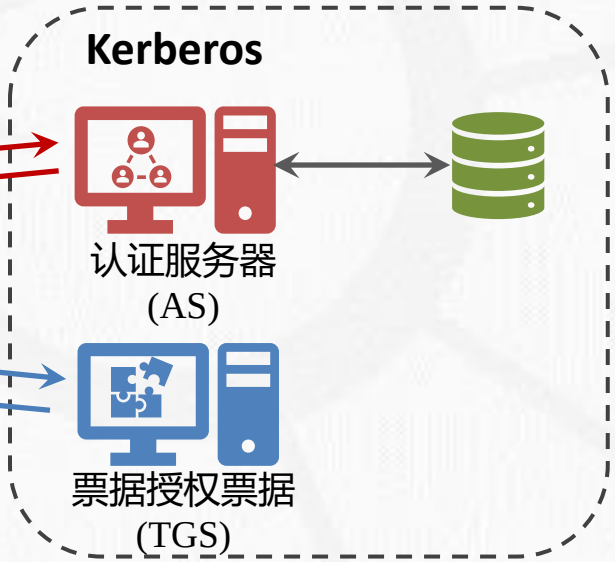
每个服务对话执行一次

请求服务

提供服务器认证符

5.工作站向服务器发送票据和认证符

2.AS在数据库中检查用户的访问权限，生活才能票据授权票据和会话密钥。用由用户口令派生出的密钥加密结果



4.TGS解密票据和认证符，验证请求，然后生成对应被请求服务器的票据

6.服务器验证票据和认证符是否匹配，然后允许访问服务。如果需要以向认证，服务器就返回一个认证符

§2.2.1 Kerberos - 简化的Kerberos v4(1)

- Kerberos v4概览

  用于获取票据授权票据的认证服务交换：

  – (1) C→AS: $ID_C||ID_{TGS}||TS_1$

  – (2) AS→C: $E_{K_C}(K_{C,TGS}||ID_{TGS}||TS_2||Lifetime_2||Ticket_{TGS})$

    - 其中K_C是由P_C生成的
    - $K_{C,TGS}$是用于客户和TGS间的本次会话密钥
    - $Ticket_{TGS} = E_{K_{TGS}}(K_{C,TGS}||ID_C||AD_C||ID_{TGS}||TS_2||Lifetime_2)$
    - TS_1可以验证客户时钟是否与AS时钟同步

C获得$K_{C,TGS}$

§2.2.1 Kerberos - 简化的Kerberos v4(2)

- Kerberos v4概览

用于获得服务授权票据的票据授权服务：

– (3) C→TGS: $ID_{SS}||Ticket_{TGS}||Authenticator_{C,TGS}$

– (4) TGS→C: $E_{K_{C,TGS}}(K_{C,SS}||ID_{SS}||TS_4||Ticket_{SS})$

C获得$K_{C,SS}$

- $K_{C,SS}$是用于客户C和业务服务器SS间的本次会话密钥
- $Ticket_{TGS} = E_{K_{TGS}}(K_{C,TGS}||ID_C||AD_C||ID_{TGS}||TS_2||Lifetime_2)$

TGS获得$K_{C,TGS}$

- $Ticket_{SS} = E_{K_{SS}}(K_{C,SS}||ID_C||AD_C||ID_{SS}||TS_4||Lifetime_4)$
- $Authenticator_{C,TGS} = E_{K_{C,TGS}}(ID_C||AD_C||TS_3)$
- TS_1可以验证客户时钟是否与AS时钟同步

§2.2.1 Kerberos - 简化的Kerberos v4(3)

- Kerberos v4概览

  为获得业务服务：

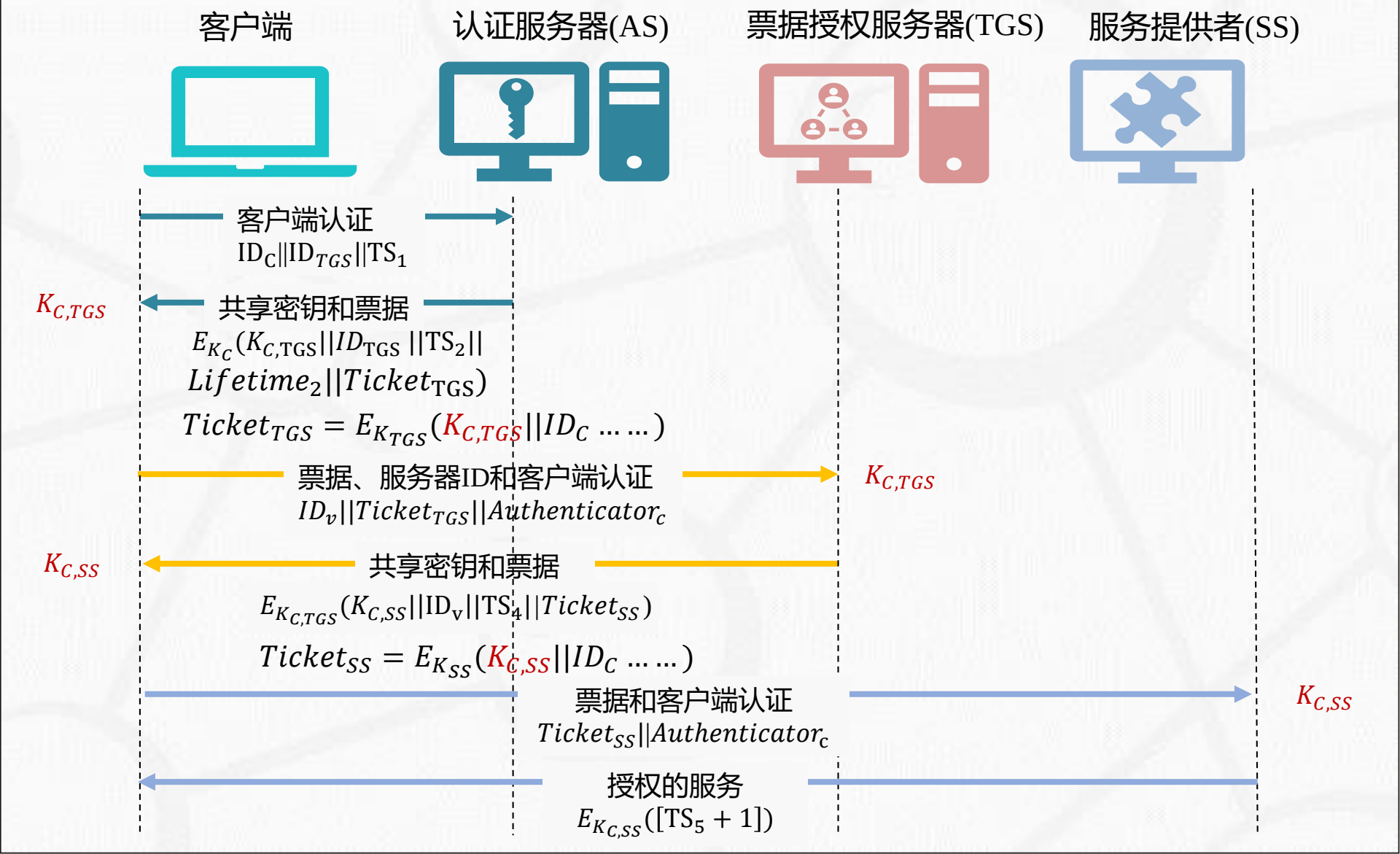
  – (5) C→SS: $Ticket_{SS}||Authenticator_{C,SS}$

  – (6) SS→C: $E_{K_{C,SS}}(TS_5+1)$

  SS获得$K_{C,SS}$

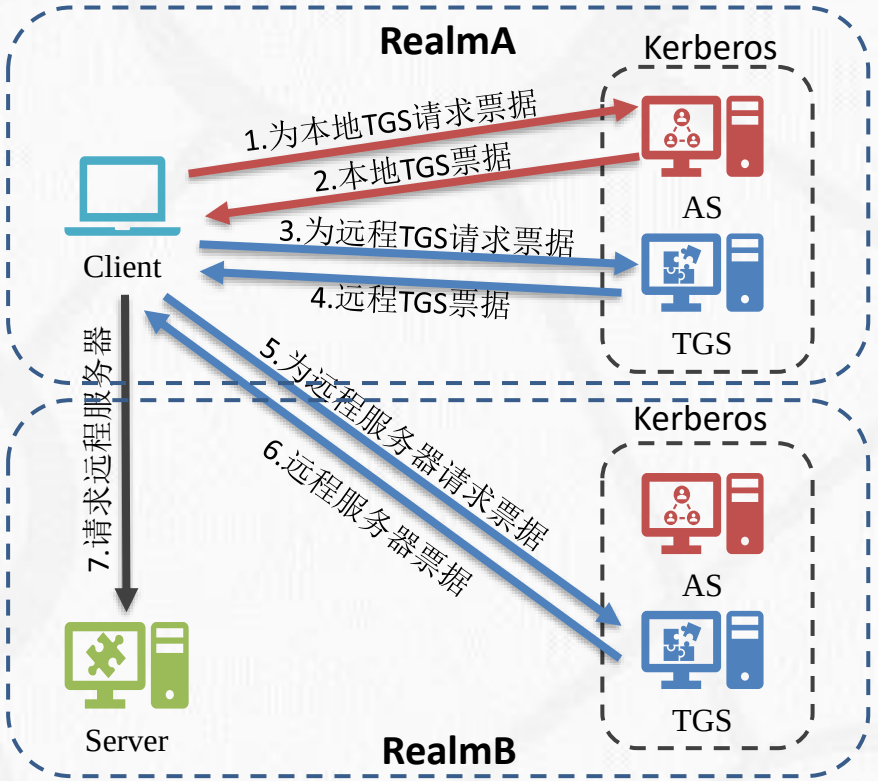
  - $Ticket_{SS}=E_{K_{SS}}(K_{C,SS}||ID_C||AD_C||ID_{SS}||TS_4||Lifetime_4)$
  - $Authenticator_{C,SS}=E_{K_{C,SS}}(ID_C||AD_C||TS_5)$
  - TS_5用于防止重放攻击

§2.2.1 Kerberos - 简化Kerberos v4的交换过程



§2.2.1 Kerberos - 关于Kerberos v4的几点说明

- Kerberos域：一个提供全套服务的Kerberos环境
  - Kerberos服务器的数据库中必须存有所有参与用户的ID和经过散列函数处理过的口令。所有用户都要在Kerberos服务器上注册。
  - Kerberos服务器必须和每个业务服务器共享一个秘密密钥。所有的业务服务器都要在Kerberos服务器上注册。
- Kerberi：
  - 多个跨域访问的Kerberos域



§2.2.1 Kerberos - v4存在的不足

- Kerberos v4存在的不足：
  - 仅能采用DES加密
    - 不兼容其它加密方法。
  - 面向互联网协议(IP)地址。
    - 不兼容其它网络协议。
  - 有效期$Lifetime$由8比特的值进行编码
    - 票据有效时间过短。假定5分钟为一个单位，最长有效期$2^8 \times 5 =$ 1280分钟，即21小时多一点。
  - 其它问题：
    - 消息字节排序、认证转发、域间认证、双重加密、PCBC加密、会话密钥、口令攻击等。

§2.2.1 Kerberos - 基本的Kerberos v5(1)

- Kerberos v5概览

用于获取票据授权票据的认证服务交换：

– (1) C→AS: $Options||ID_C||Realm_C||ID_{TGS}||Times||Nonce_1$

– (2) AS→C: $Realm_C||ID_C||Ticket_{TGS}||E_{K_C}(K_{C,TGS}||Times||Nonce_1||Realm_{TGS}||ID_{TGS})$

- $Ticket_{TGS} = E_{K_{TGS}}(Flags||K_{C,TGS}||Realm_C||ID_C||AD_C||Times)$

– 说明：

- Options：　用于参数设置，包括加密方法等。
- Realm：　用于标示客户的Kerberos域。
- Times：　票据的起止时间或者重新更新的时间。
- Nonce：　随机数。
- Flags：　根据Options设定协议的各项参数。

[1] J T Kohl, B C Neuman, T Y Ts'o. The Evolution of the Kerberos Authentication service [C]. In Distributed Open Systems. 1994.
[2] J T Kohl, B C Neuman. RFC 1510: The Kerberos Network Authentication Service (V5). 1993.

§2.2.1 Kerberos - 基本的Kerberos v5(2)

- Kerberos v5概览

用于获取服务收钱票据的票据授权服务交换：

– (3) C→TGS: $Options||ID_{SS}||Times||Nonce_2||Ticket_{TGS}||Authenticator_{C,TGS}$

– (4) TGS→C: $Realm_C||ID_C||Ticket_{SS}||E_{K_{C,TGS}}(K_{C,SS}||Times||Nonce_2||Realm_{SS}||ID_{SS})$

- $Ticket_{TGS} = E_{K_{TGS}}(Flags||K_{C,TGS}||Realm_C||ID_C||AD_C||Times)$
- $Ticket_{SS} = E_{K_{SS}}(Flags||K_{C,SS}||Realm_C||ID_C||AD_C||Times)$
- $Authenticator_{C,TGS} = E_{K_{C,SS}}(ID_C||Realm_C||TS_1)$

§2.2.1 Kerberos - 基本的Kerberos v5(3)

- Kerberos v5概览

  用于业务服务获取：

  – (5) C→SS: $Options||Ticket_{SS}||Authenticator_{C,SS}$

  – (6) SS→C: $E_{K_{C,SS}}(TS_2||Subkey||Seq\#)$

    - $Ticket_{SS} = E_{K_{SS}}(Flags||K_{C,SS}||Realm_C||ID_C||AD_C||Times)$
    - $Authenticator_{C,SS} = E_{K_{C,V}}(ID_C||Realm_C||TS_2||Subkey||Seq\#)$

  – 说明：

    - Subkey：子密钥是用户选择用来保护此次特定服务会话的加密密钥。如果忽略这个域，则使用票据中的会话密钥$K_{C,SS}$。
    - Seq#：序列号是一个可选的域。它指定了在此次会话中，服务器向客户端发送信息中所用序列号的起始值。消息可能会顺序编号以检测重放。

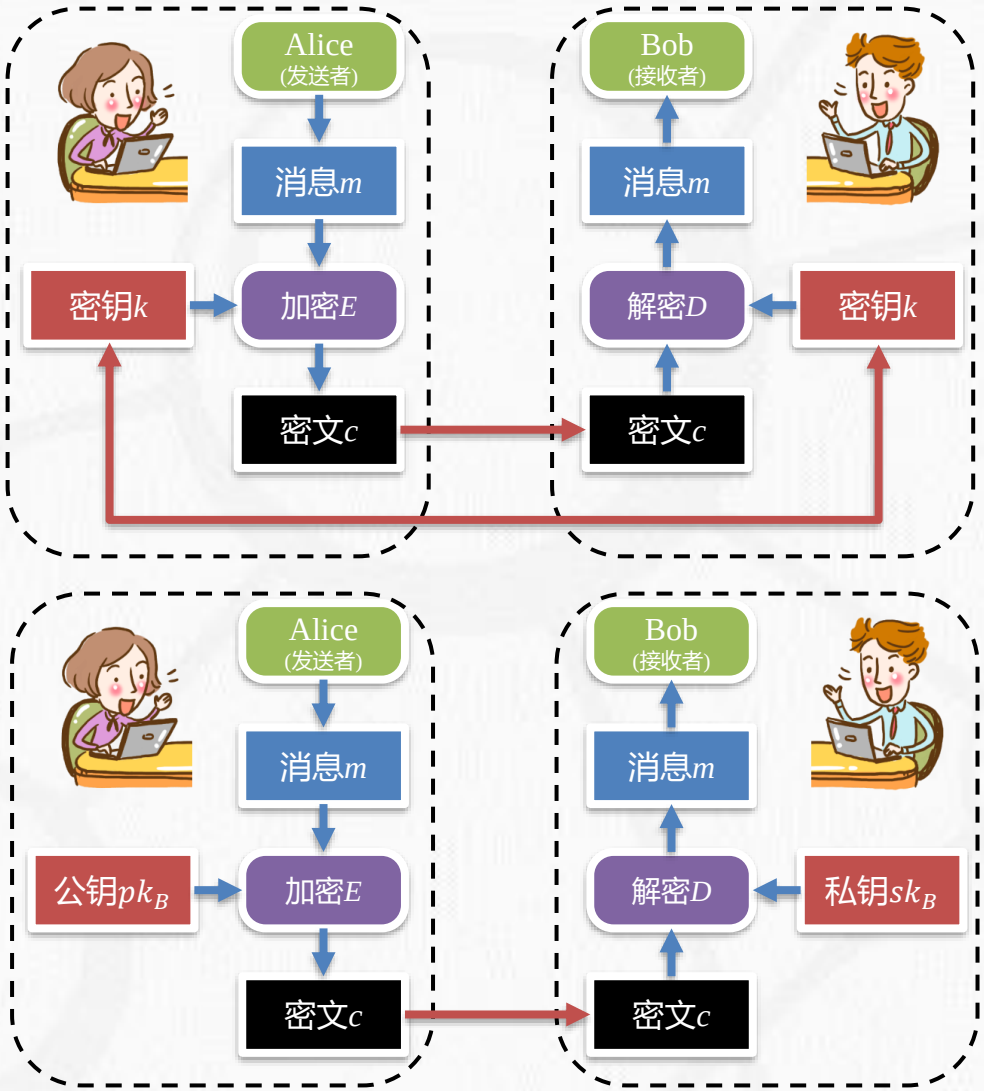
西安电子科技大学
XIDIAN UNIVERSITY

非对称加密、散列函数、数字签名

§2.2.2 X.509证书

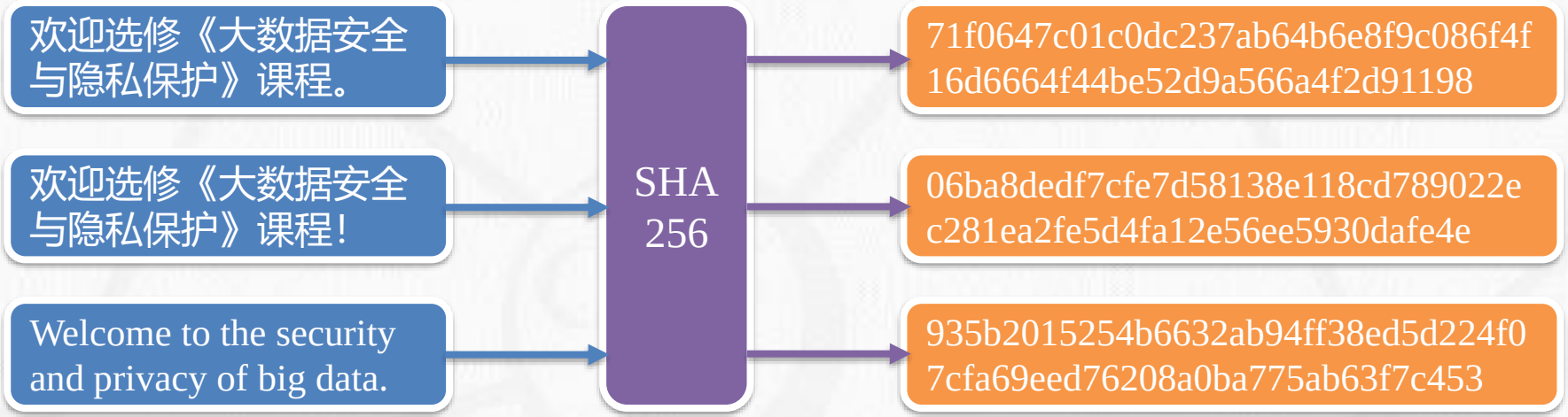
§2.2.2 X.509证书 - 密码学回顾(1)

- 对称密码：
  - 加密和解密采用相同密钥。
  - 密钥提前通过安全信道分发。
  - 加解密效率高。
  - 用途：会话加密。
- 非对称密码：
  - 加解密密钥不同。
  - 无需通过安全信道分发密钥。
  - 加解密效率低。
  - 用途：会话密钥加密。



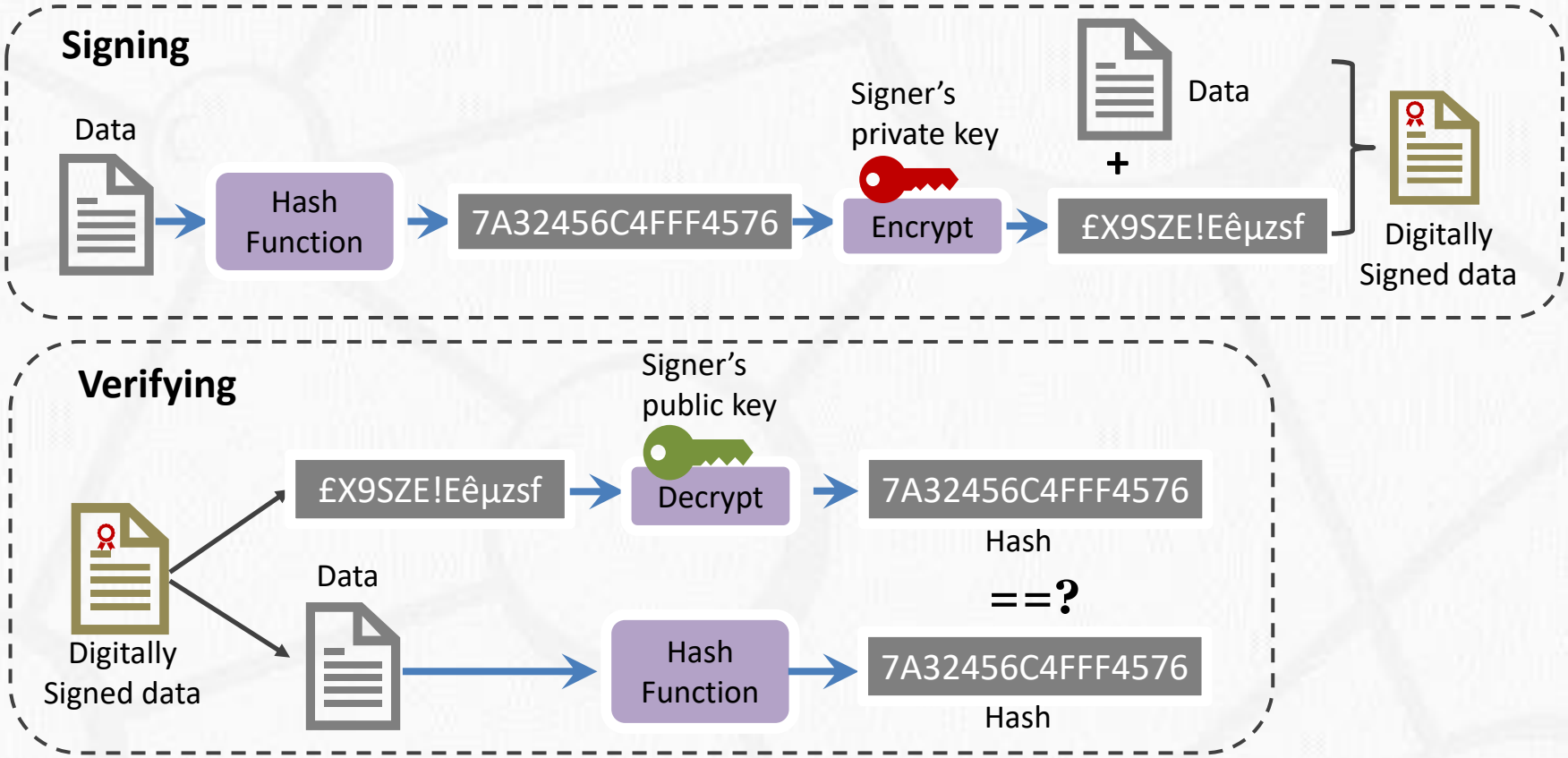
§2.2.2 X.509证书 - 密码学回顾(2)

- 散列函数：
  - 将任意长度的明文映射到固定长度的校验码。
  - 不同的明文必然映射到不同的校验码。
  - 用途：验证数据(证书、传输内容、存储内容等)的完整性。



§2.2.2 X.509证书 - 密码学回顾(3)

- 数字签名：用于保障数据(公钥证书)的认证性和完整性
  - 非对称密码用于保障数据的认证性
  - 散列函数用于保障数据的完整性



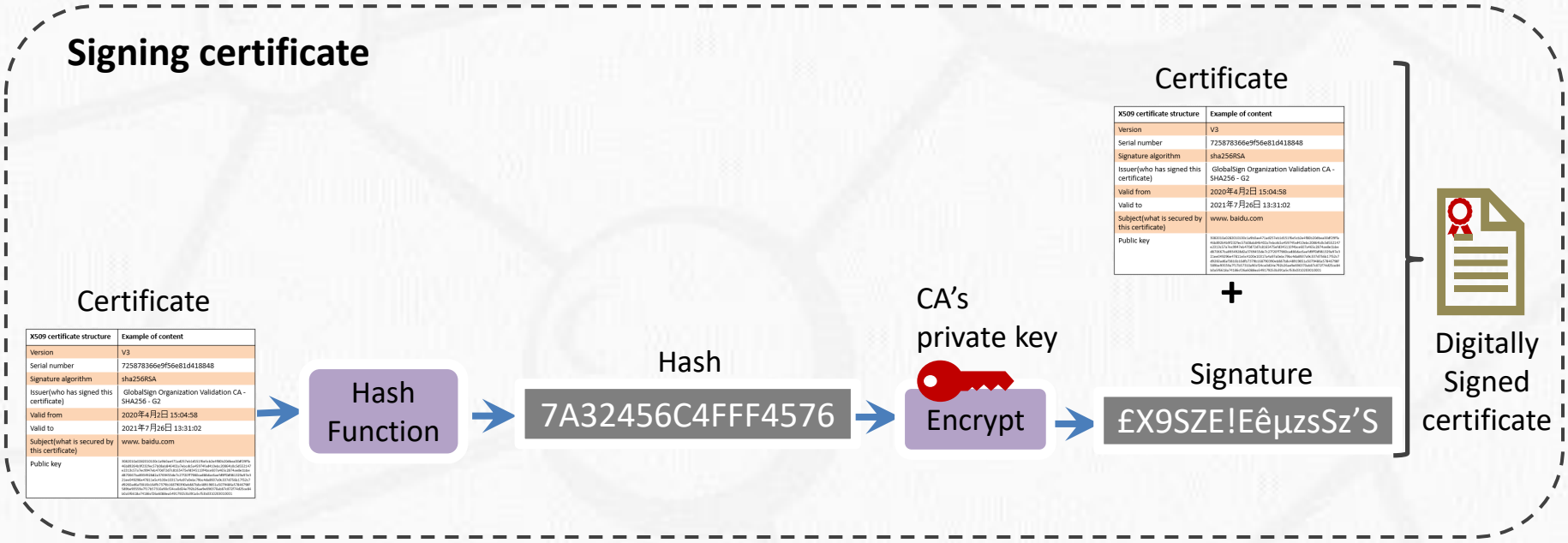
§2.2.2 X.509证书 - 概览

- 数字证书：用于认证公钥证书的从属关系。
  - 数字证书上面附带一个数字签名，用于证明这个数字证书确实为某个认证中心(CA: Certification Authority) 发布。

X509 certificate structure	Example of content
Version	V3
Serial number	725878366e9f56e81d418848
Signature algorithm	sha256RSA
Issuer(who has signed this certificate)	GlobalSign Organization Validation CA - SHA256 - G2
Valid from	02 April 2020 15:04:58
Valid to	26 July 2021 13:31:02
Subject(what is secured by this certificate)	www. baidu.com
Public key	3082010a0282010100c1a9b0ae471ad257eb1d151f6e5cb2e4f80b20dbea00df29ffa46b89264b9f232fec57b08ab846402a7ebcdc5a45974fad410ebc20864b0c5d552147e2313c57a7ec9947eb470d72d7c8165475efd345110f4bce607a465c2874ae8e1bbed870667ba8934928d2a3769455de7c27f20ff7980cad86dac6aefd9ff0d981329a97e321ee049296e47811e5c4100e10317a4a97a0ebc79bc4da8937a9c337d756b17f52c7d9260ad6af3816b16dfb7379b168790390eb887b8c48919851a5079486a57846798f589be93559a7f17b57310a90cf24ce0d24e792b26ae9e696370ab87c872f74d25ce84b0a5f6618a74186cf26a6088ea549179253b391a5cf53b0310203010001

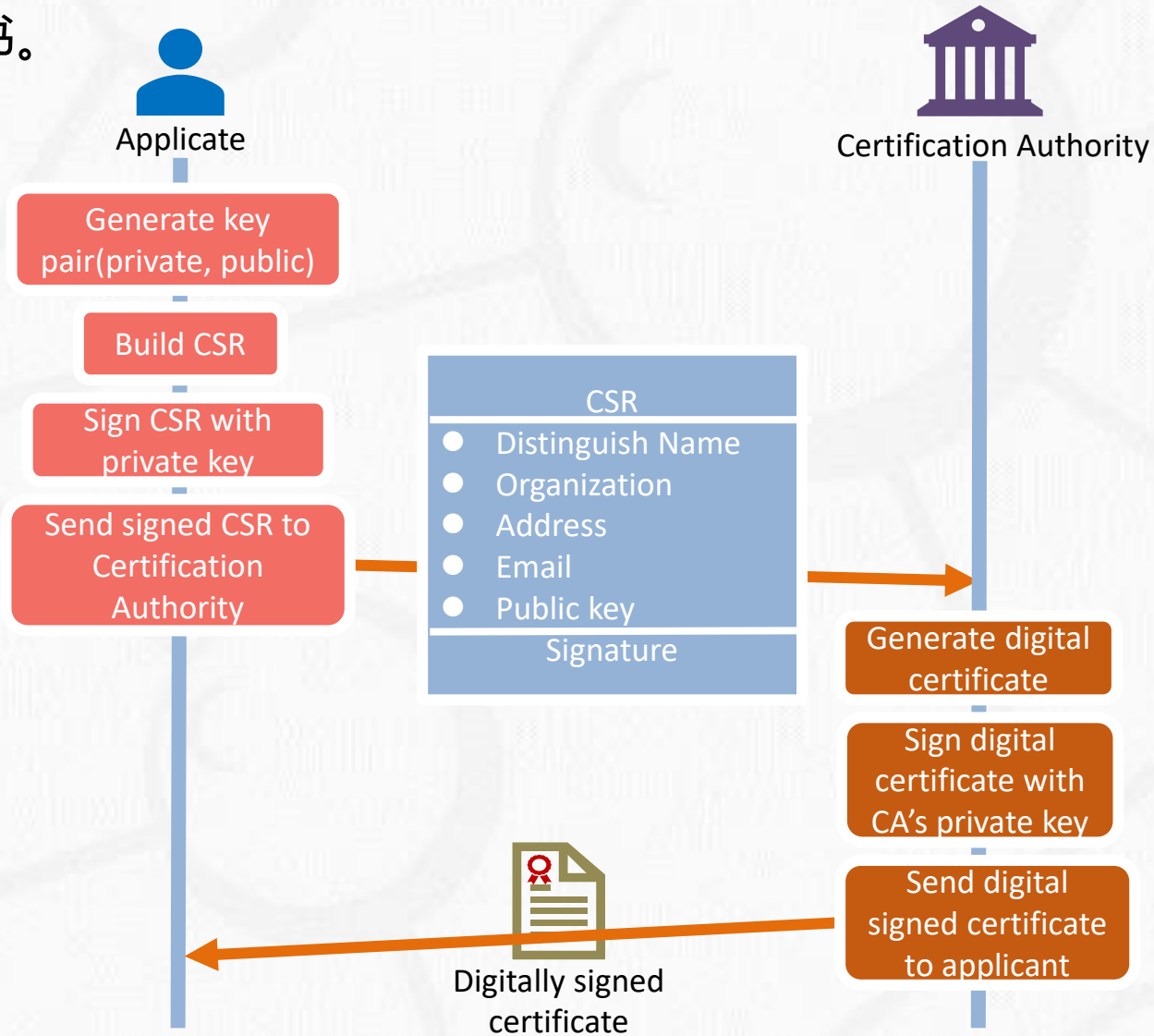
§2.2.2 X.509证书 - 认证中心CA

- 认证中心CA：
  - 向授权用户(个人)颁发数字证书；
  - 证书使用CA的私钥进行签名；
  - CA的私钥必须严格保密，这是CA作为可信第三方的基石。



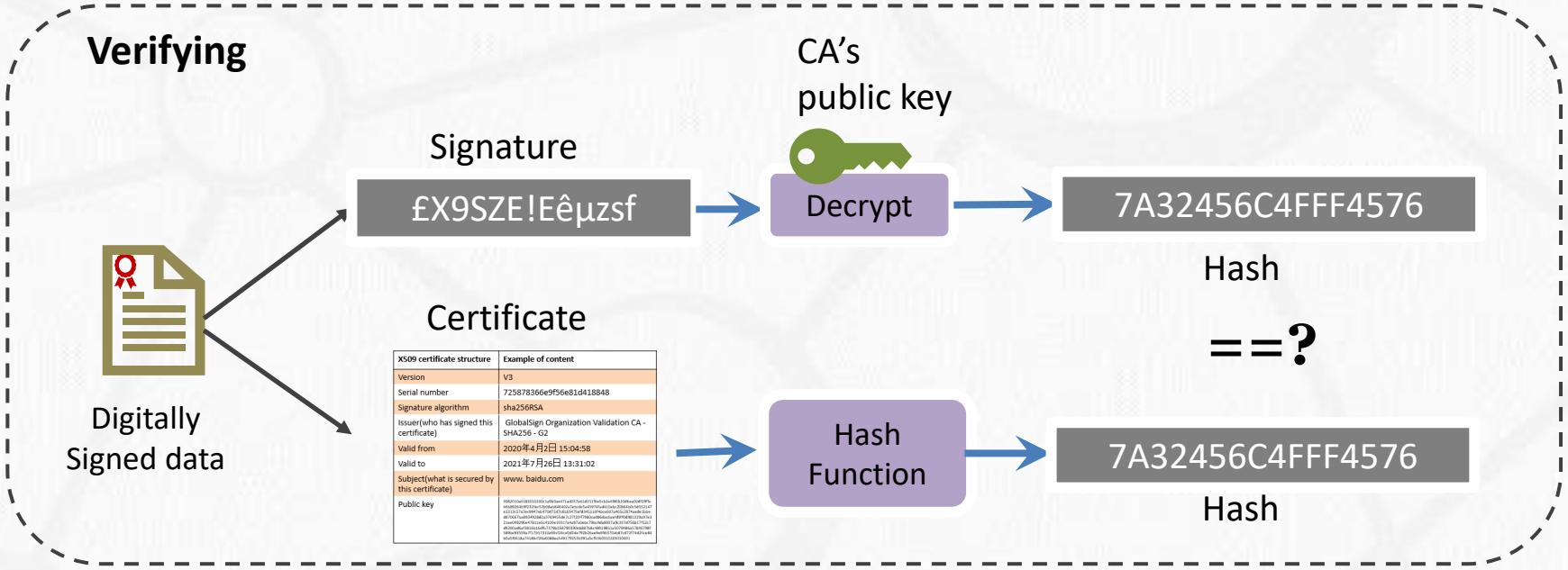
§2.2.2 X.509证书 - 证书签名请求

- 证书签名请求(CSR: Certificate Signing Request)：用于客户向CA请求一个签名的数字证书。



§2.2.2 X.509证书 - 验证数字证书

- 验证者使用CA的公钥将数字证书的签名进行解密，并与证书中的哈希值进行比对。若哈希值相等，则证书有效；否则，证书无效。



§2.2.2 X.509证书 - 证书链

- 数字证书可以通过信任链进行验证。
- 信任链：
  - 每一个公钥证书都包含自己的颁发者
  - 颁发者的公钥证书也包含他自己的颁发者
  - 由颁发者构成的这条链就是一个证书链。

X509 certificate structure	Example of content
Version	V3
Serial number	040000000001154b5ac394
Signature algorithm	sha1RSA
Issuer(who has signed this certificate)	GlobalSign Root CA
Valid from	1998年9月1日 20:00:00
Valid to	2028年1月28日 20:00:00
Subject(what is secured by this certificate)	GlobalSign Root CA
Public key	3082010a0282010100c1a9b0ae471ad257eb1d151f6e5cb2e4f80b20dbea00df29ffa46b89264b9f232fec57b08ab846402a7ebcdc5a45974fad410ebc20864b0c5d552147e2313c57a7ec9947eb470d72d7c816547 5efd3451108f4bce607a465c2874ae8e1bbed870667ba8934928d2a3769455de7c27f20ff7980cad86dac6aef69ff0d981329a97e321ee049296e47811e5c4100e10317a4a97a0ebc79bc4da8937a9c33747560a27f52c7d9260ad6af3816b16cdfb7379b168790390eb887b8c4891985 1a507948 6a57846798f5898e93559a7f17b57310a90cf24ca6bd24e792b26ae9e0637 0a8b7c872f74d25ce84b0a5f6618a74186cf26a608 8ea549179253b391a5cf53b0310203010001

X509 certificate structure	Example of content
Version	V3
Serial number	040000000001444ef04247
Signature algorithm	sha256RSA
Issuer(who has signed this certificate)	GlobalSign Root CA
Valid from	2014年2月20日 18:00:00
Valid to	2024年2月20日 18:00:00
Subject(what is secured by this certificate)	GlobalSign Organization Validation CA - SHA256 - G2
Public key	3082010a0282010100c1a9b0ae471ad257eb1d151f6e5cb2e4f80b20dbea00df29ffa46b89264b9f232fec57b08ab846402a7ebcdc5a45974fad410ebc20864b0c5d552147e2313c57a7ec9947eb470d72d7c816547 5efd3451108f4bce607a465c2874ae8e1bbed870667ba8934928d2a3769455de7c27f20ff7980cad86dac6aef69ff0d981329a97e321ee049296e47811e5c4100e10317a4a97a0ebc79bc4da8937a9c33747560a27f52c7d9260ad6af3816b16cdfb7379b168790390eb887b8c4891985 1a507948 6a57846798f5898e93559a7f17b57310a90cf24ca6bd24e792b26ae9e0637 0a8b7c872f74d25ce84b0a5f6618a74186cf26a608 8ea549179253b391a5cf53b0310203010001

X509 certificate structure	Example of content
Version	V3
Serial number	725878366e9f56e81d418848
Signature algorithm	sha256RSA
Issuer(who has signed this certificate)	GlobalSign Organization Validation CA - SHA256 - G2
Valid from	2020年4月2日 15:04:58
Valid to	2021年7月26日 13:31:02
Subject(what is secured by this certificate)	www. baidu.com
Public key	3082010a0282010100c1a9b0ae471ad257eb1d151f6e5cb2e4f80b20dbea00df29ffa46b89264b9f232fec57b08ab846402a7ebcdc5a45974fad410ebc20864b0c5d552147e2313c57a7ec9947eb470d72d7c816547 5efd3451108f4bce607a465c2874ae8e1bbed870667ba8934928d2a3769455de7c27f20ff7980cad86dac6aef69ff0d981329a97e321ee049296e47811e5c4100e10317a4a97a0ebc79bc4da8937a9c33747560a27f52c7d9260ad6af3816b16cdfb7379b168790390eb887b8c4891985 1a507948 6a57846798f5898e93559a7f17b57310a90cf24ca6bd24e792b26ae9e0637 0a8b7c872f74d25ce84b0a5f6618a74186cf26a608 8ea549179253b391a5cf53b0310203010001

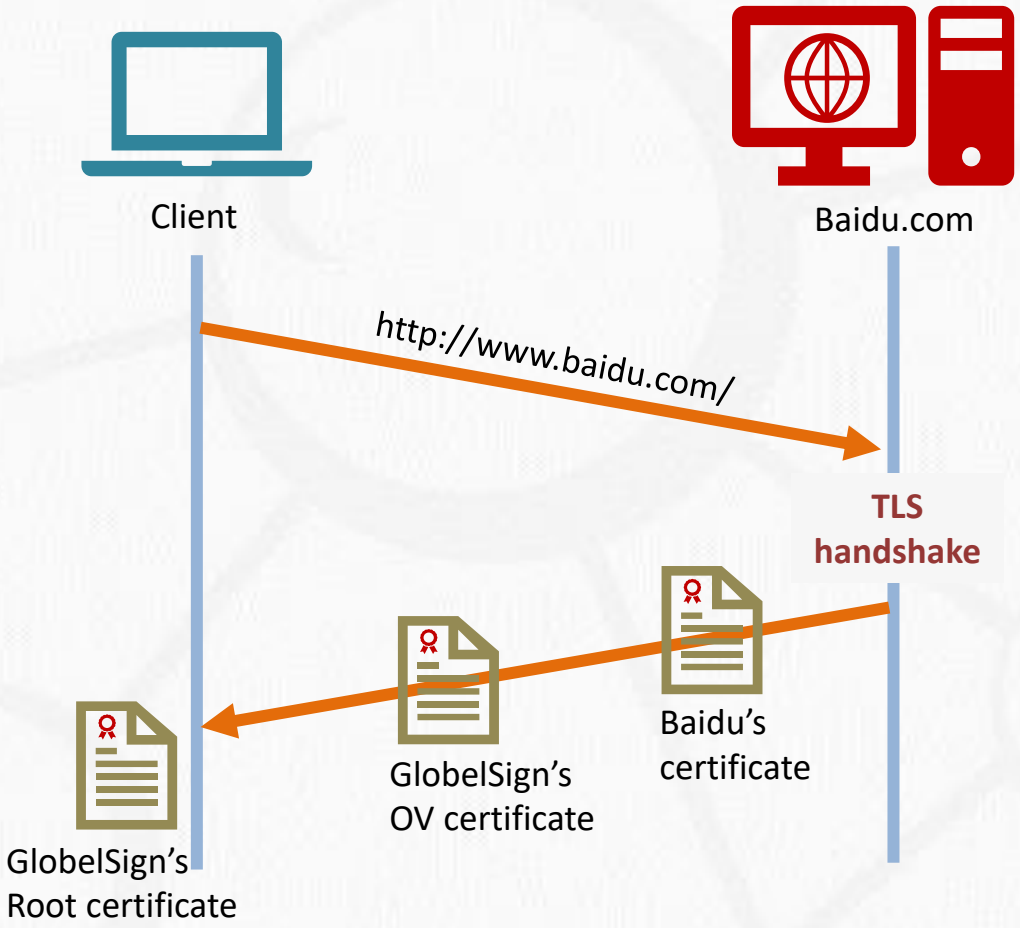
§2.2.2 X.509证书 - 根证书

- 根证书
  - 位于证书链最顶端的证书；
  - 由一个可信的CA颁发，采用自签名的形式生成；
  - 根证书往往在安全物理环境中生成。

X.509 certificate structure	Example of content
Version	V3
Serial number	040000000001154b5ac394
Signature algorithm	sha1RSA
Issuer(who has signed this certificate)	GlobalSign Root CA
Valid from	01 September 1998 20:00:00
Valid to	28 January 2028 20:00:00
Subject(what is secured by this certificate)	GlobalSign Root CA
Public key	3082010a0282010100c1a9b0ae471ad257eb1d151f6e5cb2e4f80b20dbea00df29ffa46b89264b9f232fec57b08ab846402a7ebcdc5a45974fad410ebc20864b0c5d552147e2313c57a7ec9947eb470d72d7c8165475efd345110f4bce607a465c2874ae8e1bbed870667ba8934928d2a3769455de7c27f20ff7980cad86dac6aefd9ff0d981329a97e321ee049296e47811e5c4100e10317a4a97a0ebc79bc4da8937a9c337d756b17f52c7d9260ad6af3816b16dfb7379b168790390eb887b8c48919851a5079486a57846798f589be93559a7f17b57310a90cf24ce0d24e792b26ae9e696370ab87c872f74d25ce84b0a5f6618a74186cf26a6088ea549179253b391a5cf53b0310203010001

§2.2.2 X.509证书 - 证书的验证

- 证书的验证：
  - (1) 客户(比如：浏览器)发起一个TLS连接，获得Baidu和GlobelSign's OV公钥数字证书；
  - (2) 客户使用GlobelSign's OV证书中的公钥去验证Baidu的证书；
  - (3) 客户使用GlobelSign's Root证书中的公钥去验证GlobelSign's OV的证书；
  - (4) 客户端已经内置存储GlobelSign's Root的根证书。



§2.2.2 X.509证书 - 证书编码

- DER vs. CRT vs. CER vs. PEM

Certificate encodings	Encoding type
PEM(Privacy Enhanced Mail)	Base64 ASCII encoding
DER(Distinguished Encoding Rules)	Binary encoding

Certificate file extension	Encoding type
.cer	PEM or DER
.crt	PEM or DER
.der	DER
.pem	PEM

DER encoded, .cer file extension, certificate

```
000  30 82 05 fb 30 82 04 e3 a0 03 02 01 02 02 10 3f 80  0...û0...ã ......?.
011  f3 13 48 22 3a db 19 b5 ac d4 c5 6c bc cb 30 0d 06  ó.H":Û.µ¬ÔÅl¼Ë0..
022  09 2a 86 48 86 f7 0d 01 01 0b 05 00 30 7e 31 0b 30  .*.H.÷......0~1.0
033  09 06 03 55 04 06 13 02 55 53 31 1d 30 1b 06 03 55  ...U....US1.0...U
044  04 0a 13 14 53 79 6d 61 6e 74 65 63 20 43 6f 72 70  ....Symantec Corp
055  6f 72 61 74 69 6f 6e 31 1f 30 1d 06 03 55 04 0b 13  oration1.0...U...
066  16 53 79 6d 61 6e 74 65 63 20 54 72 75 73 74 20 4e  .Symantec Trust N
077  65 74 77 6f 72 6b 31 2f 30 2d 06 03 55 04 03 13 26  etwork1/0-..U...&
088  53 79 6d 61 6e 74 65 63 20 43 6c 61 73 73 20 33 20  Symantec Class 3
099  53 65 63 75 72 65 20 53 65 72 76 65 72 20 43 41 20  Secure Server CA
0aa  2d 20 47 34 30 1e 17 0d 31 37 31 31 30 32 30 30 30  - G40...171102000
0bb  30 30 30 5a 17 0d 31 38 31 31 30 33 32 33 35 39 35  000Z..18110323595
0cc  39 5a 30 67 31 0b 30 09 06 03 55 04 06 13 02 55 53  9Z0g1.0...U....US
0dd  31 13 30 11 06 03 55 04 08 0c 0a 57 61 73 68 69 6e  1.0...U....washin
0ee  67 74 6f 6e 31 10 30 0e 06 03 55 04 07 0c 07 53 65  gton1.0...U....Se
0ff  61 74 74 6c 65 31 19 30 17 06 03 55 04 0a 0c 10 41  attle1.0...U....A
110  6d 61 7a 6f 6e 2e 63 6f 6d 2c 20 49 6e 63 2e 31 16  mazon.com, Inc.1.
121  30 14 06 03 55 04 03 0c 0d 77 77 77 2e 61 6d 61 7a  0...U....www.amaz
132  6f 6e 2e 66 72 30 82 01 22 30 0d 06 09 2a 86 48 86  on.fr0.."0...*.H.
143  f7 0d 01 01 01 05 00 03 82 01 0f 00 30 82 01 0a 02  ÷...........0....
```

PEM encoded, .pem file extension, certificate

```
000  2d 2d 2d 2d 2d 42 45 47 49 4e 20 43 45 52 54 49 46  -----BEGIN CERTIF
011  49 43 41 54 45 2d 2d 2d 2d 2d 0a 4d 49 49 44 76 44  ICATE-----.MIIDvD
022  43 43 41 79 57 67 41 77 49 42 41 67 49 4a 41 4d 62  CCAywgAwIBAgIJAMb
033  48 42 41 6d 38 49 6c 75 67 4d 41 30 47 43 53 71 47  HBAm8IlugMA0GCSqG
044  53 49 62 33 44 51 45 42 42 51 55 41 4d 49 47 62 4d  SIb3DQEBBQUAMIGbM
055  51 73 77 43 51 59 44 0a 56 51 51 47 45 77 4a 4b 55  QswCQYD.VQQGEwJKU
066  44 45 4f 4d 41 77 47 41 31 55 45 43 42 4d 46 56 47  DEOMAwGA1UECBMFVG
077  39 72 65 57 38 78 45 44 41 4f 42 67 4e 56 42 41 63  9reW8xEDAOBgNVBAc
088  54 42 30 4e 6f 64 57 38 74 61 33 55 78 45 54 41 50  TB0Nodw8ta3UxETAP
099  42 67 4e 56 0a 42 41 6f 54 43 45 5a 79 59 57 35 72  BgNV.BAoTCEZyYw5r
0aa  4e 45 52 45 4d 52 67 77 46 67 59 44 56 51 51 4c 45  NEREMRgwFgYDVQQLE
0bb  77 39 58 5a 57 4a 44 5a 58 4a 30 49 46 4e 31 63 48  w9XZwJDZXJ0IFN1cH
0cc  42 76 63 6e 51 78 47 44 41 57 42 67 4e 56 42 41 4d  BvcnQxGDAWBgNVBAM
0dd  54 0a 44 30 5a 79 59 57 35 72 4e 45 52 45 49 46 64  T.D0ZyYw5rNEREIFd
0ee  6c 59 69 42 44 51 54 45 6a 4d 43 45 47 43 53 71 47  lYiBDQTEjMCEGCSqG
0ff  53 49 62 33 44 51 45 4a 41 52 59 55 63 33 56 77 63  SIb3DQEJARYUc3Vwc
110  47 39 79 64 45 42 6d 63 6d 46 75 61 7a 52 6b 0a 5a  G9ydEBmcmFuazRk.Z
121  43 35 6a 62 32 30 77 48 68 63 4e 4d 44 63 78 4d 6a  C5jb20wHhcNMDcxMj
132  41 33 4d 54 41 79 4d 54 51 32 57 68 63 4e 4d 54 63  A3MTAyMTQ2WhcNMTc
143  78 4d 6a 41 30 4d 54 41 79 4d 54 51 32 57 6a 43 42  xMjA0MTAyMTQ2WjCB
```

§2.2.2 X.509证书 - 证书仓库

- 证书仓库(Certificate store)：一般内建在操作系统或浏览器中
  - 存储自身的私钥
  - 存储自身的证书
  - 可信第三方的证书
  - 证书撤销列表
- 大多数知名的根证书都存储在证书仓库中。
- 用户可以根据实际需要在证书仓库中添加根证书。

⚠ 仅仅在证书仓库中添加可信证书！⚠

§2.2.2 X.509证书 - 过期证书

- 证书有效期：
  - 每个证书都有一个固定的有效期，超过有效期证书将失效
  - 在证书过期之前，应当申请一个新的证书

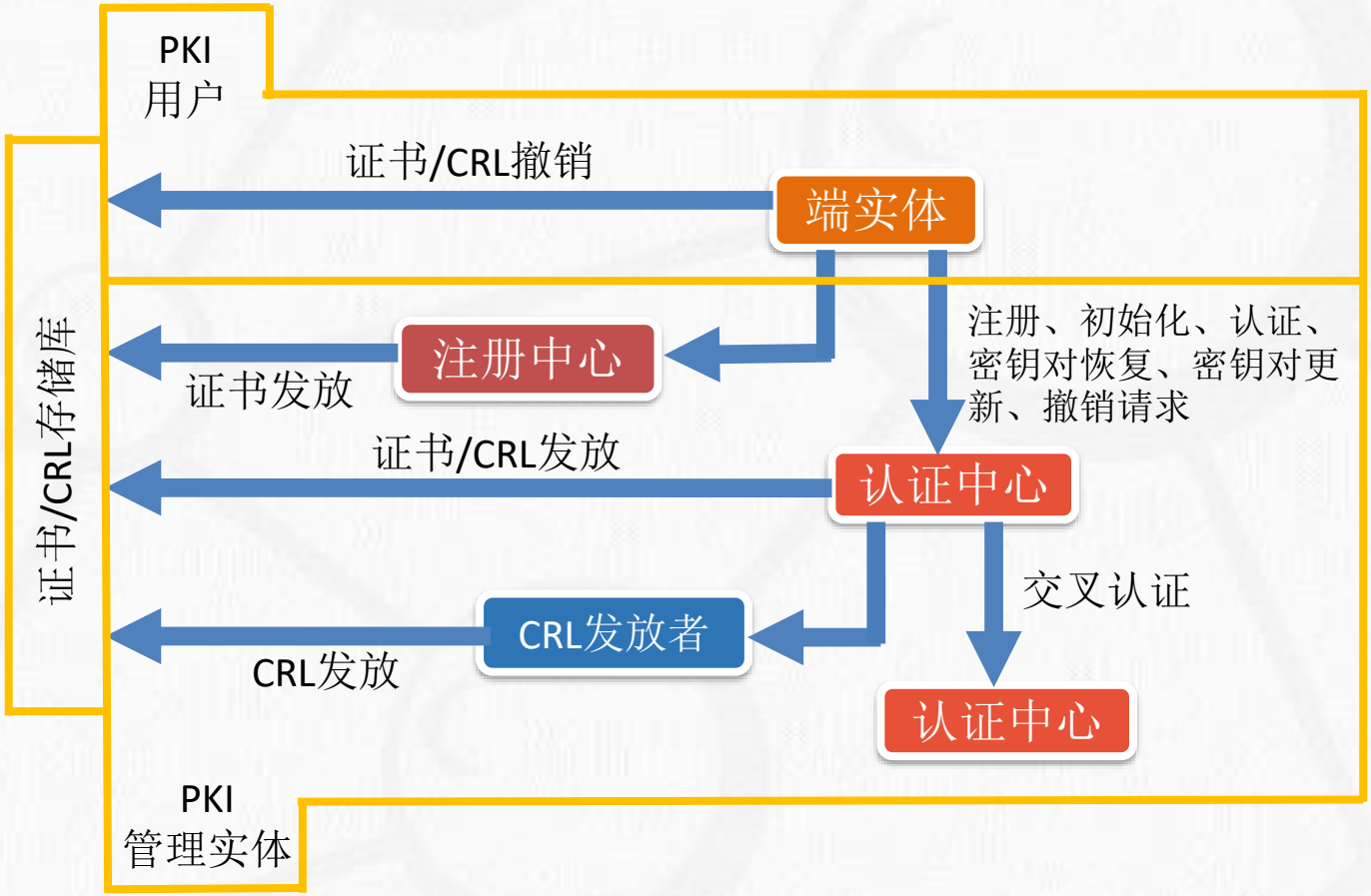
- 证书撤销列表：
  - 存放有效期内作废的公钥证书
    - 私钥丢失、被窃等
  - 存在于证书撤销列表内的证书视为无效证书
  - 由根证书机构进行维护

§2.2.2 X.509证书 - 公钥基础设施PKI

- 公钥基础设施(PKI: Public Key Infrastructure)
  - **端实体**：一个用来表示终端用户、设备（比如服务器和路由器）或者任何其他的可以在公钥证书的主体域被确定身份的实体的通用术语。端实体一般采用和/或支持与PKI相关的服务。
  - **认证中心(CA)**： 证书的发放者，通常也是撤销证书列表（CRL）的发放者。它还可能支持很多管理功能，虽然这些一般是由一个或多个注册中心代理的。
  - **注册中心(RA)**：一个可选的部分，它承担很多从CA处继承的管理功能。经常将RA与端实体注册过程关联起来，但是也可以协助许多其他领域的工作。
  - **撤销证书列表(CRL)**：一个可选的部分，它可以代理CA发布CRL。一个用来表示储存证书和CRL以使证书和CRL可以被端实体检索的任何方法的通用术语。
  - **存储库**：一个用来表示储存证书和CRL以使证书和CRL可以被端实体检索的任何方法的通用术语。

§2.2.2 X.509证书 - 公钥基础设施PKI

- 公钥基础设施(PKI: Public Key Infrastructure)



§2.2.2 X.509证书 – PKI管理功能

- PKI管理功能：
  - **注册**：这是一个过程，用户通过该过程在CA向其发放证书（一个或几个）之前，先让CA知道自己（直接或通过RA）。注册开始了一个PKI中的登记过程。注册通常包括一些离线或在线的步骤来互相认证。一般来说，为将来认证使用的一个和多个共享密钥被发放给端实体。
  - **初始化**：在客户端可以安全工作以前，需要安装密钥资料，这些密钥资料与存储在基础设施其他地方的密钥具有一定的关系。例如，客户端需要被安全地初始化，这需要使用公钥以及其他由可信CA（一个或多个）担保的、将被用于验证证书路径的信息。
  - **认证**：这是一个过程，在此过程中，CA为一个用户的公钥发放一个证书，并将该证书返回给用户的客户端系统和/或将此证书存放在一个储存库中。

§2.2.2 X.509证书 – PKI管理功能

- PKI管理功能：
  - **密钥对恢复**：密钥对可用来支持数字签名的产生和验证，或者可用来支持加密和解密，或者两者都支持。如果一个密钥是用来加密和解密的，那么当不再能以通常的方式访问密钥资料时，提供一种机制来恢复解密密钥就是非常重要的，否则，就不可能恢复加密的数据。不能访问解密密钥，可能由以下情况导致：忘记口令/PIN码、磁盘驱动损坏、硬件标记损坏等。密钥对恢复功能允许端实体从一个被授权的密钥备份设施（通常是给端实体发放证书的CA）处恢复他们的加密/解密密钥对。
  - **密钥对更新**：所有密钥对都需要定期更新（例如用一个新的密钥对代替），并且发放新证书。当证书过期或证书被撤销时，就需要更新。

§2.2.2 X.509证书 – PKI管理功能

- PKI管理功能：
  - **撤销申请**：一个经过授权的人告诉CA发生了一个异常情况，需要撤销证书。撤销的原因包括私钥泄露，合作方变化和名称改变。
  - **交叉认证**：两个CA互相交换用于建立交叉证书信息。一个交叉证书是一个CA给另一个CA发放的证书，证书中包含一个CA用于发放证书的签名密钥。

§1 小结

- 内容回顾
  - Kerberos的工作流程，Kerberos v4和v5的主要区别
  - X.509证书的生成流程、组成要素、验证过程
  - PKI的实体组成、工作流程

- 掌握
  - PKI的基本架构和实体任务
  - X.509证书的一般生成步骤和验证步骤
  - Kerberos协议中各关键字段的含义

Thanks!
Questions & Advices!