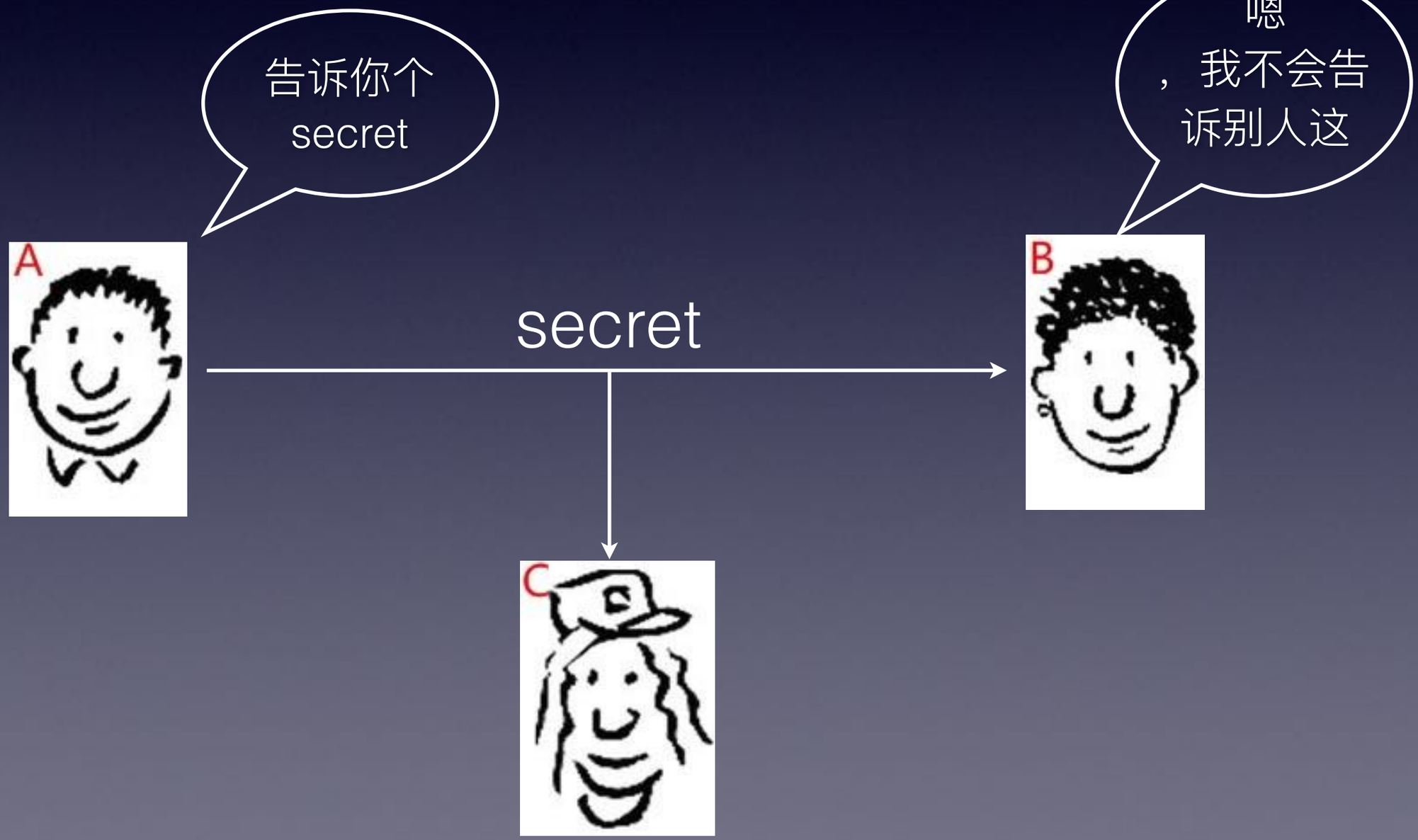
SSL/TLS

Karos Li
You dream it, we build it.

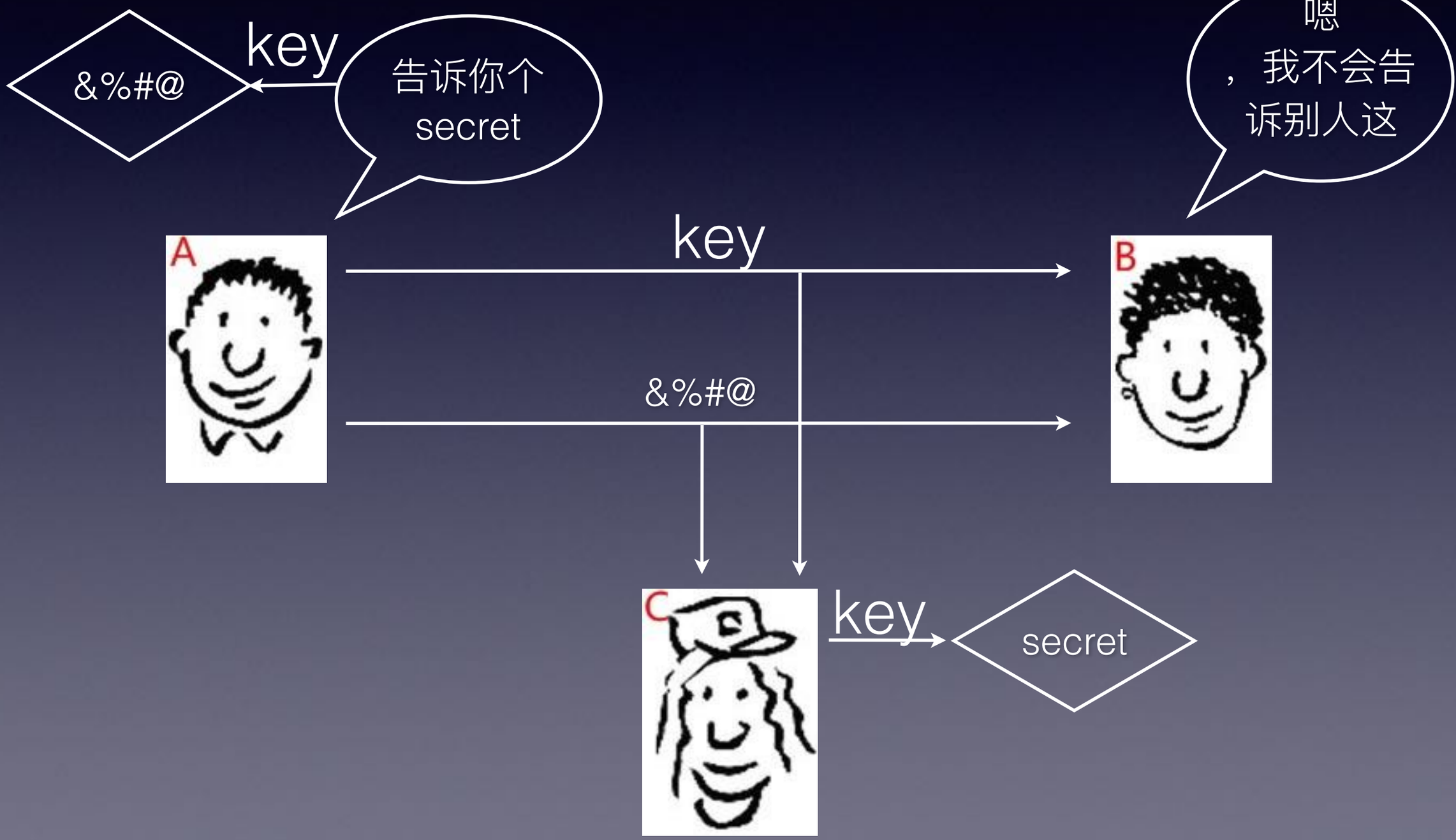
Agenda

- Insecurity cases
- Background
- Evolution of SSL
- SSL Work Process

Plain Text



Cipher Text



Background

基于万维网的电子商务和网上银行等新兴应用，极大地方便了人们的日常生活，受到人们的青睐。由于这些应用都需要在网络上进行在线交易，它们对网络通信的安全性提出了更高的要求。传统的万维网协议HTTP不具备安全机制——采用明文的形式传输数据、不能验证通信双方的身份、无法防止传输的数据被篡改等，导致HTTP无法满足电子商务和网上银行等应用的安全性要求。

三大风险：

- 窃听风险（eavesdropping）：第三方可以获知通信内容。
- 篡改风险（tampering）：第三方可以修改通信内容。
- 冒充风险（pretending）：第三方可以冒充他人身份参与通信。

Background

- SSL(Secure Socket Layer)是netscape公司设计的主要用以保障在Internet上数据传输安全，利用数据加密(Encryption)技术，可确保数据在网络上之传输过程中不会被截取及窃听。一般通用之规格为40 bit之安全标准，美国则已推出128 bit之更高安全标准，但限制出境。当前版本为3.0。它已被广泛地用于Web浏览器与服务器之间的身份认证和加密数据传输。

- IETF(www.ietf.org)将SSL作了标准化，即RFC2246,并将其称为TLS（Transport Layer Security），从技术上讲，TLS1.0与SSL3.0的差别非常微小。

- 在WAP的环境下，由于手机及手持设备的处理和存储能力有限，wap论坛（www.wapforum.org）在TLS的基础上做了简化，提出了WTLS协议（Wireless Transport Layer Security），以适应无线的特殊环境。

对称加密 symmetric cryptographic

- 分组密码是将明文按一定的位长分组，明文组经过加密运算得到密文组，密文组经过解密运算（加密运算的逆运算），还原成明文组，典型的例子有DES,RC5,IDEA。

- 序列密码是指利用少量的密钥（制乱元素）通过某种复杂的运算（密码算法）产生大量的伪随机位流，用于对明文位流的加密。解密是指用同样的密钥和密码算法及与加密相同的伪随机位流，用以还原明文位流，典型的例子是RC4。

- CBC(Cipher Block Chaining)模式这个词在分组密码中经常会用到，它是指一个明文分组在被加密之前要与前一个的密文分组进行异或运算。当加密算法用于此模式的时候除密钥外，还需协商一个初始化向量（IV），这个IV没有实际意义，只是在第一次计算的时候需要用到而已。采用这种模式的话安全性会有所提高。

- 简单的说就是加密和解密用的同一个密钥。

- 优点：加解密速度快。缺点：容易暴露密钥。

- E(msg, key) = emsg。 D(emsg,key)=msg。

非对称加密 asymmetric cryptographic

- 是指一对加密密钥与解密密钥，这两个密钥是数学相关，用某用户密钥加密后所得的信息，只能用该用户的解密密钥才能解密。如果知道了其中一个，并不能计算出另外一个。因此如果公开了一对密钥中的一个，并不会危害到另外一个的秘密性质。称公开的密钥为公钥；不公开的密钥为私钥。

- 如果加密密钥是公开的，这用于客户给私钥所有者上传加密的数据，这被称作为公开密钥加密(狭义)。例如，网络银行的客户发给银行网站的账户操作的加密数据。

- 简单的说就是加密密钥与解密密钥不同，分私钥和公钥。这种方法大多用于密钥交换，RSA便是一个我们熟知的例子。

- 优点：安全性高，不用暴露私钥。缺点：加解密速度慢。

- E(msg, pkey)=emsg。 D(emsg, pkey)!=msg。 D(emsg, skey)=msg。

Evolution of SSL -- Public Key

告诉你一个秘密

#&%@

A用B的公钥加密一个消息并告诉B，B通过私钥能够识别这个消息，说明B就是A所要找的人。

#&%@

告诉你一个秘密

SecretKey

#&%@1

A生成会话密钥，并用B的公钥加密。B通过私钥得到会话密钥。

#&%@1

SecretKey

A和B高兴私聊中。

如果只是提供公钥和私钥，任何人都可以是B，只要跟A谎称自己就是那个B，然后提供公钥代替B的公钥，就可以用自己的私钥来证明自己的身份，这样A就不能分辨出你不是B。

Evolution of SSL -- Public Key

告诉你一个秘密

#&%@

A用B*的公钥加密消息并告诉B*，B*通过私钥能够识别这个消息，说明B*就是A所要找的人。

#&%@

告诉你一个秘密

SecretKey

#&%@1

A生成会话密钥，并用B*的公钥加密。B*通过私钥得到会话密钥。

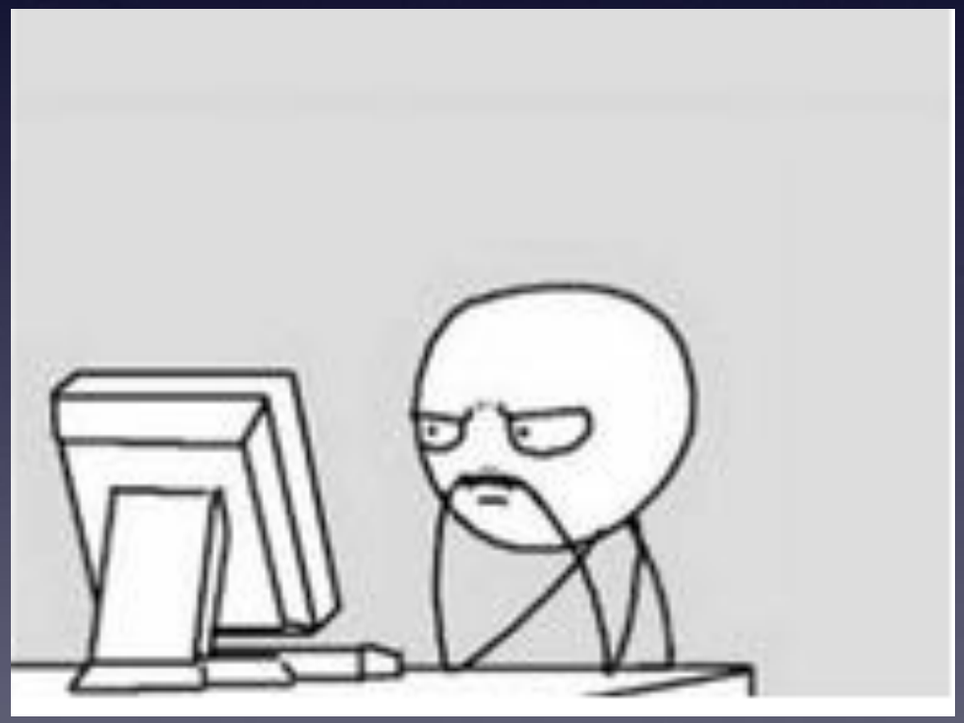
#&%@1

SecretKey

A和B*高兴私聊中。

Why not private chat with me?

纳尼！



Evolution of SSL -- Certificate

- 数字证书就是互联网通讯中标志通讯各方身份信息的一串数字，提供了一种在Internet上验证通信实体身份的方式，其作用类似于司机的驾驶执照或日常生活中的身份证。它是由一个由权威机构-----CA机构，又称为证书授权（Certificate Authority）中心发行的，人们可以在网上用它来识别对方的身份。数字证书是一个经证书授权中心数字签名的包含公开密钥拥有者信息以及公开密钥的文件。最简单的证书包含一个公开密钥、名称以及证书授权中心的数字签名。
- 目前数字证书的格式普遍采用的是X.509V3国际标准，一个标准的X.509数字证书包含以下一些内容：

Evolution of SSL -- Certificate

- CA机构，又称为证书授证（Certificate Authority）中心，作为电子商务交易中受信任的第三方，承担公钥体系中公钥的合法性检验的责任。CA中心为每个使用公开密钥的用户发放一个数字证书，数字证书的作用是证明证书中列出的用户合法拥有证书中列出的公开密钥。CA机构的数字签名使得攻击者不能伪造和篡改证书。它负责产生、分配并管理所有参与网上交易的个体所需的数字证书，因此是安全电子交易的核心环节。由此可见，建设证书授权（CA）中心，是开拓和规范电子商务市场必不可少的一步。为保证用户之间在网上传递信息的安全性、真实性、可靠性、完整性和不可抵赖性，不仅需要对用户的身份真实性进行验证，也需要有一个具有权威性、公正性、唯一性的机构，负责向电子商务的各个主体颁发并管理符合国内、国际安全电子交易协议标准的电子商务安全证书。

- 数字证书颁发过程一般为：管理员只需生成“证书请求”（后缀大多为.csr），它包含你的个人信息和公钥，然后把这份请求交给诸如Globlesign，verisign等有CA服务公司（当然，连同几百美金），你的证书请求经验证后，CA用它的私钥签名，形成正式的证书发还给你。管理员再在server上导入这个证书就行了。

- 怎么验证证书的有效性：当客户端收到服务[illegible]将收到的证书与本地系统的根证书（第三方CA的证书，内置在系统中）进行对比，如果[illegible]中某个CA签发的，就意味着服务器端是真实有效的。

Evolution of SSL -- Certificate

- Self-signed certificate：由自己的私钥签名而非第三方CA签发的证书。如果你不想花那笔钱，可以自己做CA，现在有一些工具（openssl，keynote）可以帮助生成自定义的CA，服务器的证书和自签名。当服务器端安装好有自定义CA签名的证书时，这个时候客户端是需要导入自定义的CA证书，意味着客户端“信任”这个CA签署的证书）。而商业CA的一般不用，因为它们已经内置在系统中了。

Evolution of SSL -- Certificate

CA验证

B给A发送自己的证书。A验证B的证书，然后从证书中取得公钥。

SecretKey

#&%@1

A生成会话密钥，并用B的公钥加密。B通过私钥得到会话密钥。

#&%@1

SecretKey

A和B又高兴私聊中。

由于公钥加密只能协商出一个会话密钥，通过会话密钥可以保证通信的机密性。但是不能保证通信的完整性。

Evolution of SSL -- Certificate

CA验证

B给A发送自己的证书。A验证B的证书，然后从证书中取得公钥。

A生成会话密钥，并用B的公钥加密。B通过私钥得到会话密钥。

SecretKey

#&%@1

#&%@1

SecretKey

我喜欢你 我去年买了个表

SecretKey

#&%@2 我去年买了个

B用会话密钥加密消息并发给A。

#&%@2

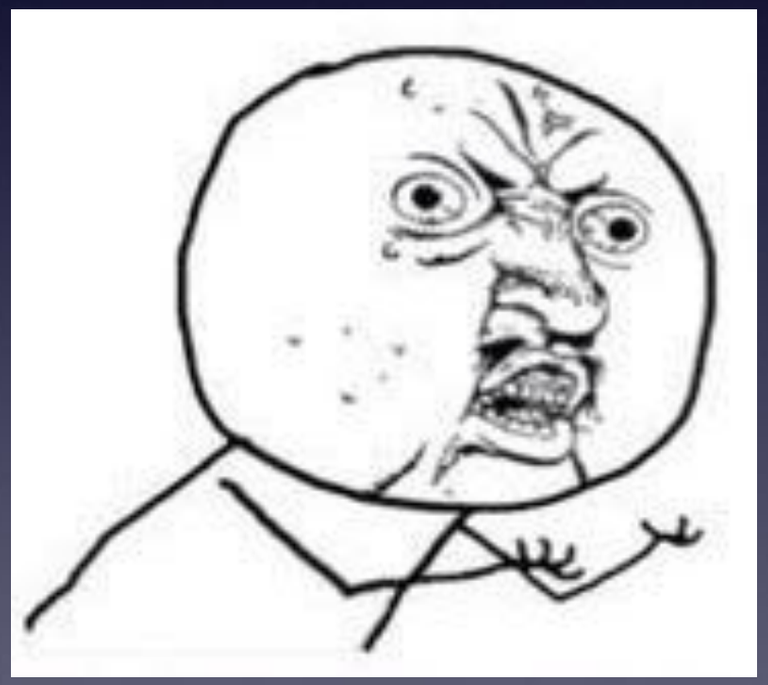
SecretKey

我喜欢你

#&%@2 我去年买了个表

#&%@2

玛莎卡！



Evolution of SSL -- Signature

- 数字签名是指用原文进行HASH运算得到摘要信息并用发送者的私钥加密，与原文一起传送给接收者。接收者只有用发送者的公钥才能解密被加密的摘要信息，然后用HASH函数对收到的原文产生一个摘要信息，与解密的摘要信息对比。如果相同，则说明收到的信息是完整的，在传输过程中没有被修改，否则说明信息被修改过，因此数字签名能够验证信息的完整性。

发送方：

接收方：

哈希算法 Hash

- 单向散列函数又称单向Hash函数，就是把任意长的输入消息串变化成固定长的输出串且由输出串难以得到输入串的一种函数。这个输出串称为该消息的散列值。一般用于产生消息摘要，密钥加密等.

- Hash函数主要用于完整性校验和提高数字签名的有效性,目前已有很多方案。这些算法都是伪随机函数,任何杂凑值都是等可能的。输出并不以可辨别的方式依赖于输入;在任何输入串中单个比特的变化,将会导致输出比特串中大约一半的比特发生变化。
  常见单向散列函数(Hash函数)
  ·MD5（Message Digest AIgorithm 5）：是RSA数据安全公司开发的一种单向散列算法，MD5被广泛使用，可以用来把不同长度的数据块进行暗码运算成一个128位的摘要；
  ·SHA（Secure Hash AIgorithm）这是一种较新的散列算法，可以对任意长度的数据运算生成一个160位的摘要，比128位散列更能有效抵抗穷举攻击；
  ·MAC（Message Authentication Code）：消息认证代码，是一种使用密钥的单向函数，可以用它们在系统上或用户之间认证文件或消息。HMAC（用于消息认证的密钥散列法）就是这种函数的一个例子。

- 由于单向散列的算法都是公开的，所以其它人可以先改动原文，再生成另外一份摘要。解决这个问题的办法可以通过HMAC（RFC 2104）,它包含了一个密钥，只有拥有相同密钥的人才能鉴别这个散列。

- E(msg,key)=emsg=E(msg,key)。 D(emsg,key)!=msg。

Evolution of SSL -- Signature

CA验证

B给A发送自己的证书。A验证B的证书，然后从证书中取得公钥。

A生成会话密钥，并用B的公钥加密。B通过私钥得到会话密钥。

SecretKey

#&%@1

#&%@1

SecretKey

Digest

SecretKey
HASH

#&%@2

B用会话密钥加密消息并发给A。

#&%@2

SecretKey

我喜欢你

Digest

SecretKey
HASH

Encrypted Signature

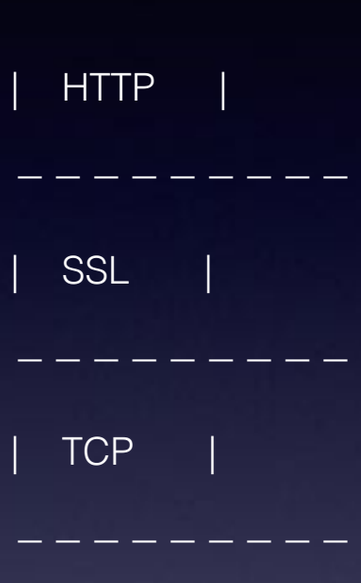
Encrypted Signature

SecretKey
HASH

A和B再次高兴的私聊中

SSL Hierarchy

SSL是一个介于HTTP协议与TCP之间的一个可选层，其位置大致如下：



如果利用SSL协议来访问网页，其步骤如下：

用户：在浏览器的地址栏里输入https://www.sslserver.com

HTTP层：将用户需求翻译成HTTP请求，如

GET /index.htm HTTP/1.1

Host www.sslserver.com

SSL Hierarchy

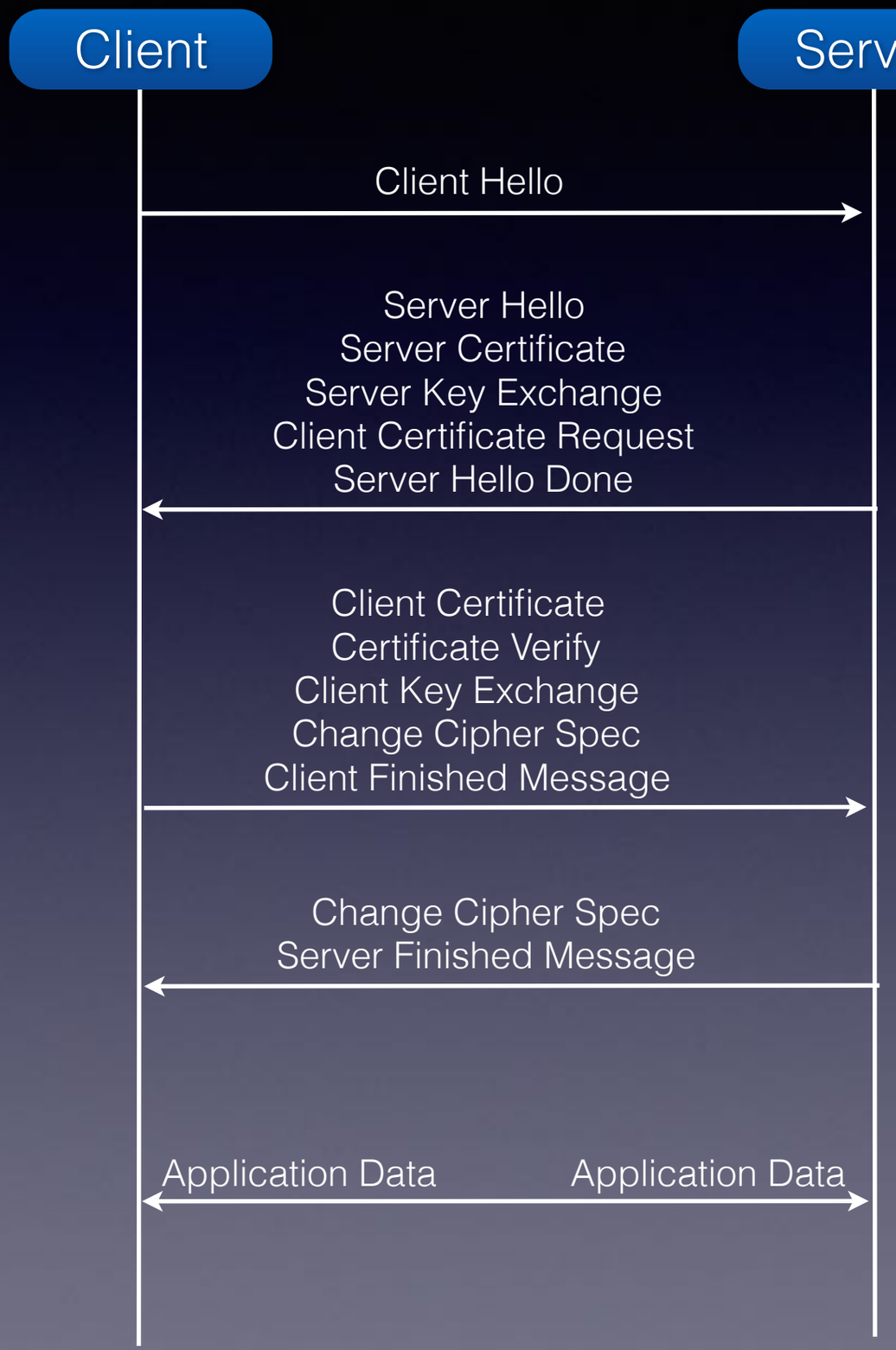
SSL层: 借助SSL层协议的的信道安全的协商出一份加密密钥，并用此密钥来加密HTTP请求。

TCP层：与web server的443端口建立连接，传递SSL处理后的数据。

SSL在TCP之上建立了一个加密通道，通过这一层的数据经过了加密，因此达到保密的效果。

SSL协议可分为部分： SSL记录协议（SSL Record Protocol）：它建立在可靠的传输协议（如TCP）之上，为高层协议提供数据封装、压缩、加密等基本功能的支持。
SSL握手协议（SSL Handshake Protocol）：它建立在SSL记录协议之上，用于在实际的数据传输开始前，通讯双方进行身份认证、协商加密算法、交换加密密钥等。

Handshake Process



1.客户（client）端发送ClientHello信息给服务器（Server）端，Server回答ServerHello。这个过程建立的安全参数包括协议版本、“会话”标识、加密算法、压缩方法。另外，还交换两个随机数：ClientHello.Random和

2.Hello消息发送完后，Server端会发送它的证书和密钥交换信息。如果Server端被认证，它就会请求Client端的证书，在验证以后，Server就发送HelloDone消息，以示达成

3.Server请求Client证书时，Client要返回证书或返回没有证书的指示，这种情况用于单向认证，即客户端不装有证书。Client发送密钥交换消息，然后产生会话密钥并通知Server,

4.Server端确认会话密钥，并向Client发送“握手完成”消息

Handshake Process

SSL中的握手协议，将公钥加密技术与对称密钥加密技术的应用有效、巧妙地结合在一起，有机地组成了互联网（或其他网络）上信息安全传输的通道。这种信息安全通道，有其实用价值，比如，利用对称加密技术比公钥加密技术对大容量信息的加/解密速度要快，而公钥技术却提供了更好的身份认证技术。SSL的握手协议可以非常有效地让客户与服务器之间完成身份认证。

通过SSL客户端与服务器传送自己的数字证书，互验合法性，特别是验证服务器的合法性，可以有效地防止互联网上虚假网站的网上钓鱼事件；同时，服务器端也可以严格验证客户端的真实身份。其作用如下：

① 客户端的浏览器向服务器传送客户端SSL协议的版本号、加密算法的种类、产生的随机数，以及其他服务器和客户端之间通信所需要的各种信息。
② 服务器向客户端传送SSL协议的版本号、加密算法的种类、随机数及其他相关信息，同时，服务器还将向客户端传送自己的证书。
③ 客户利用服务器传过来的信息验证服务器的合法性。服务器的合法性包括：证书是否过期，发行服务器证书的CA是否可靠，发行者证书的公钥能否正确解开服务器证书的“发行者的数字签名”，服务器证书上的域名是否和服务器的实际域名相匹配。如果合法性验证没有通过，则通信将断开；如果合法性验证通过，则将继续进行第④步
④ 客户端随机产生一个用于后面通信的“对称密码”，然后用服务器的公钥（从步骤②中服务器的证书中获得）对其加密，再将加密后的“预主密码”传给服务器。
⑤ 如果服务器要求客户的身份认证（在握手过程中为可选），用户则可以建立一个随机数，然后对其进行数字签名，将这个含有签名的随机数和客户自己的证书，以及加密过的“预主密码”一起传给服务器。

Handshake Process

⑥ 如果服务器要求客户的身份认证，服务器则必须检验客户证书和签名随机数的合法性。具体的合法性验证包括：客户的证书使用日期是否有效，为客户提供证书的CA是否可靠，发行CA的公钥能否正确解开客户证书的发行CA的数字签名，检查客户的证书是否在证书撤销列表（CRL）中。检验如果没有通过，则通信立刻中断；如果验证通过，则服务器将用自己的私钥解开加密的“预主密码”，然后执行一系列步骤来产生主通信密码（客户端也将通过同样的方法产生相同的主通信密码）。

⑦ 服务器和客户端用相同的主密码，即“通话密码”，一个对称密钥用于SSL协议的安全数据通信的加/解密通信。同时，在SSL通信过程中还要完成数据通信的完整性，以防止数据通信中的任何变化。

⑧ 客户端向服务器端发出信息，指明后面的数据通信将使用步骤⑦中的主密码为对称密钥，同时通知服务器客户端的握手过程结束。

⑨ 服务器向客户端发出信息，指明后面的数据通信将使用步骤⑦中的主密码为对称密钥，同时通知客户端服务器端的握手过程结束。

⑩ SSL的握手部分结束，SSL安全通道的数据通信开始，客户和服务器开始使用相同的对称密钥进行数据通信，同时进行通信完整性的检验。

SSL Work Process

这里我们用个形象的比喻，我们假设A与B通信，A是SSL客户端，B是SSL服务器端，加密后的消息放在方括号[]里，以突出明文消息的区别。双方的处理动作的说明用圆括号（）括起。

A：我想和你安全的通话，我这里的对称加密算法有DES,RC5,密钥交换算法有RSA和DH，摘要算法有MD5和SHA。

B：我们用DES－RSA－SHA这对组合好了。这是我的证书，里面有我的名字和公钥，你拿去验证一下我的身份（把证书发给A）。目前没有别的可说的了。

A：（查看证书上B的名字是否无误，并通过手头早已有的CA的证书验证了B的证书的真实性，如果其中一项有误，发出警告并断开连接，这一步保证了B的公钥的真实性）产生一份秘密消息，这份秘密消息处理后将用作加密密钥，加密初始化向量和hmac的密钥。将这份秘密消息-协议中称为per_master_secret-用B的公钥加密，封装成称作ClientKeyExchange的消息。由于用了B的公钥，保证了第三方无法窃听）

我生成了一份秘密消息，并用你的公钥加密了，给你（把ClientKeyExchange发给B）

注意，下面我就要用加密的办法给你发消息了！

（将秘密消息进行处理，生成加密密钥，加密初始化向量和hmac的密钥）

[我说完了]

B：（用自己的私钥将ClientKeyExchange中的秘密消息解密出来，然后将秘密消息进行处理，生成加密密钥，加密初始化向量和hmac的密钥，这时双方已经安全的协商出一套加密办法了）

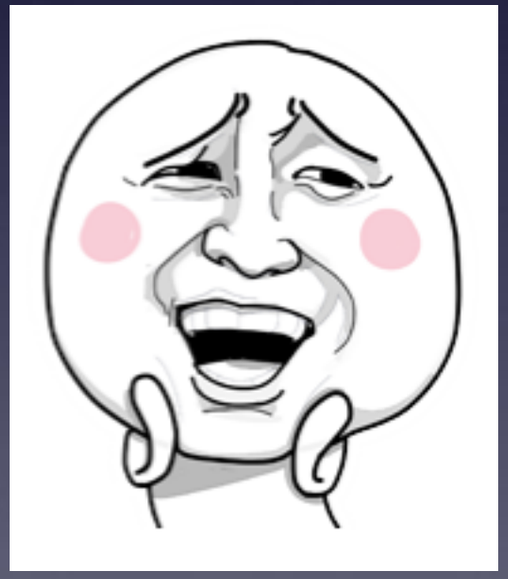
注意，我也要开始用加密的办法给你发消息了！

[我说完了]

A: [我的秘密是...]

简单的说便是：SSL客户端（也是TCP的客户端）在TCP链接建立之后，发出一个ClientHello来发起握手，这个消息里面包含了自己可实现的算法列表和其它一些需要的消息，SSL的服务器端会回应一个ServerHello，这里面确定了这次通信所需要的算法，然后发过去自己的证书（里面包含了身份和自己的公钥）。Client在收到这个消息后会生成一个秘密消息，用SSL服务器的公钥加密后传过去，SSL服务器端用自己的私钥解密后，会话密钥协商成功，双方可以用同一份会话密钥来通信了。

妈妈再也不用担心
我在线私聊了！



SSL is absolutely secure?

Reference

http://tools.ietf.org/html/rfc5246

http://3layer.blog.51cto.com/57448/20430/

http://netsecurity.51cto.com/art/201108/287971.htm

http://www.educity.cn/labs/568700.html

http://hi.baidu.com/cqs2006123/item/70af9d46d9d3153efb896049

Thank You!