2016乌云白帽大会・不插电

分析和攻击私有协议中的密码学安全漏洞

刘慧

关于我

上海交通大学密码与计算机安全实验室
Lab of Cryptology and Computer Security

应用密码学研究
现实软件中密码系统的安全审计分析

软件安全小组GoSSIP
Group of Software Security In Progress

WooYun.org

乌云白帽子/乌云专栏作者
Gossip on SSL Security by GoSSIP_SJTU

私有协议中的密码学安全漏洞

什么是协议

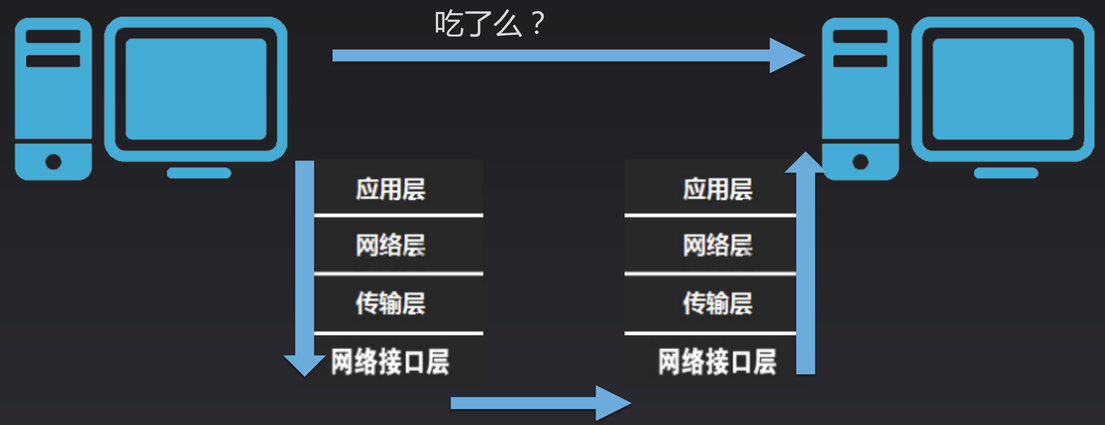
网络通信协议，为进行数据交换而建立的规则、标准或约定的集合
它规定了通信时信息必须采用的格式和这些格式的意义

应用层：
0x00 0x26 0xe5 0x90 0x83 0xe4 0xba 0x86 … 0xbc 0x9f
标志位：采用何种压缩算法，字符编码方式，长度，…

网络层：
源端口，目的端口，…

传输层：
源IP，目的IP，…

网络接口层：
MAC地址，…



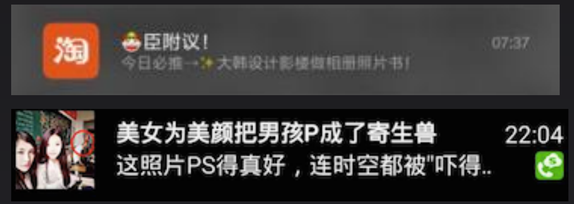
私有协议中的密码学安全漏洞

我们关心的协议

与相应实体通信并完成特定功能的应用层协议。

HTTP、FTP、SMTP、…

也可以是更加应用相关的协议，例如
- 手机APP与发送推送信息的服务器间的通信协议
- 即时消息应用间及其与服务器的通信协议
- 网络摄像头与中心服务器的通信协议
- ……



私有协议中的密码学安全漏洞

私有协议
Proprietary protocol

非标准协议
- 微信、QQ；自定义格式的数据交换
- 可能缺少公开的、详细的协议规范文档

协议逆向
- 抓包分析
- 二进制反汇编反编译

私有协议中的密码学安全漏洞

协议的关注点

对于网络摄像头

✗ 每帧画面是如何编码的，画面质量与清晰度的协商调整，…

✓ 每帧画面传输前是否有协商某些安全机制，传输的画面是否可能被截获或修改

对于即时消息

✗ 消息的某几个字节对应系统内部维护的序列号

✓ 消息能否冒充、伪造

私有协议中的密码学安全漏洞

协议的关注点

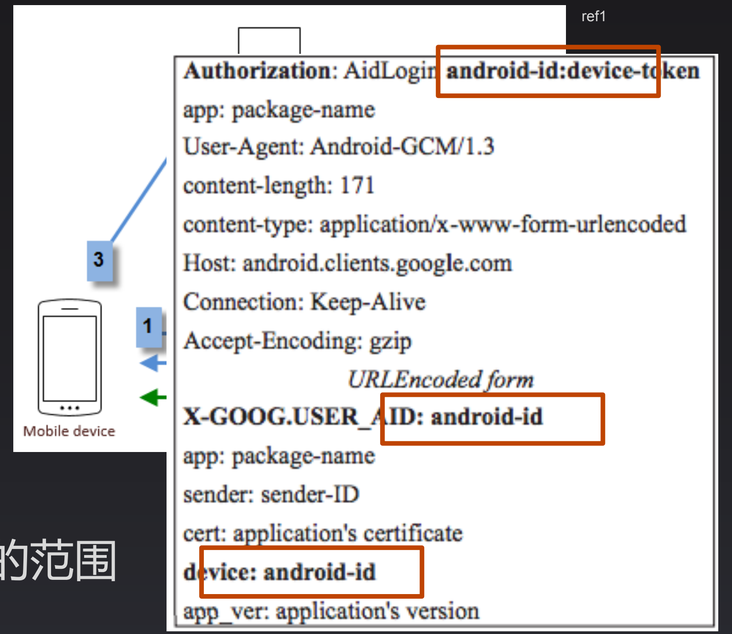
对于推送协议

✖某个字段对应标题，某个字段对应推送消息点击后跳转的网页

✔服务器推送来的消息本身是否在传输过程中可能被修改

例如，

谷歌云消息服务，曾经因服务器没有检查客户端提交的请求中两个字段的一致性，导致原本推送到客户端的消息会被攻击者收到[Ref2]。



这种协议中业务逻辑相关的安全问题不是我们讨论的范围

Ref 1:http://intown.biz/wp-content/uploads/2014/04/gcm_explained.png

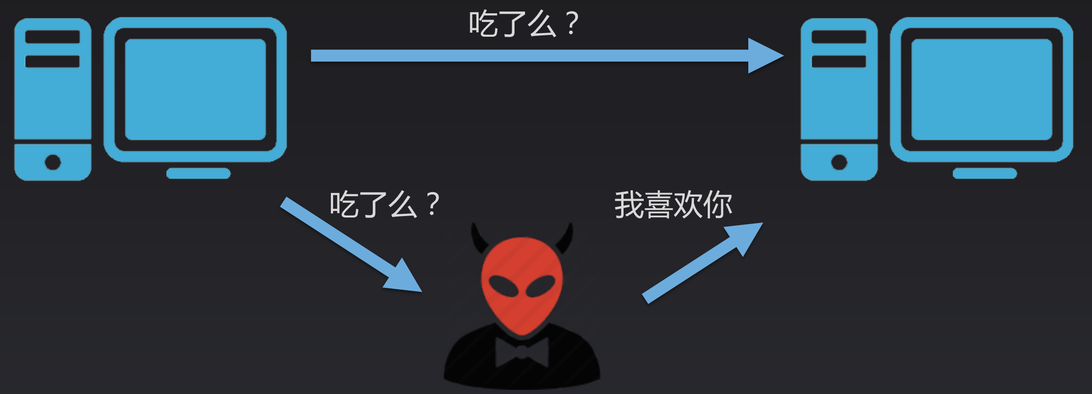
Ref2:Li, Tongxin, et al. "Mayhem in the push clouds: Understanding and mitigating security hazards in mobile push-messaging services." Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014.

私有协议中的密码学安全漏洞

What we really focus on

信息传送的通道是否安全

在存在恶意中间人的情况下，通信的安全性
- 被动中间人：窃听
- 主动中间人：篡改、重放
- 身份冒充



安全传输三大愿望

真实性：来自真实的的发送方

Authenticity

机密性：未授权不能读

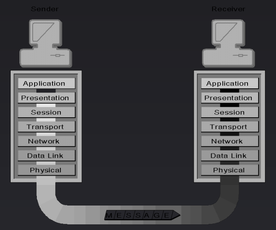
Confidentiality

完整性：未授权不能修改

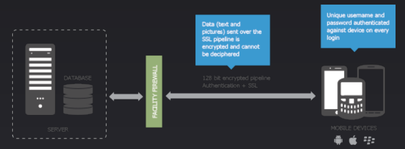
Integrity

私有协议中的密码学安全漏洞

网络通信协议
Communications Protocol



数据安全传输
Secure Data Transmission



密码学误用
Cryptographic Misuse

分析私有协议中的密码学问题时我们要关注什么

我正在跟我想要的人讲话

讲话的内容不想让别人知道

讲话的内容别人不知道也不能乱改

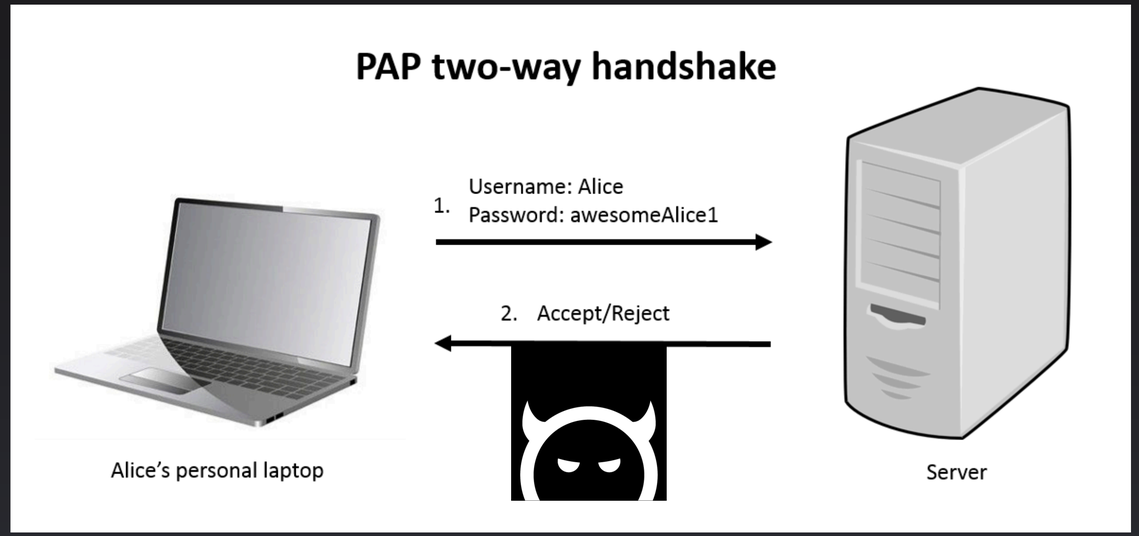
身份认证

确认通信对象身份

基于密码的身份认证

密码认证协议(PAP)

Password Authentication Protocol



基于密码的身份认证

依托下层通道的安全性

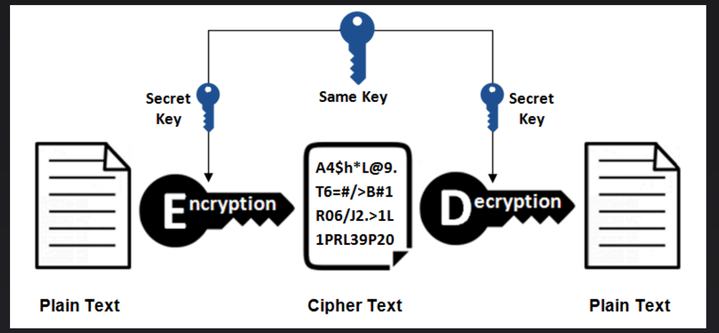
HTTPS， HTTPS with Pinning

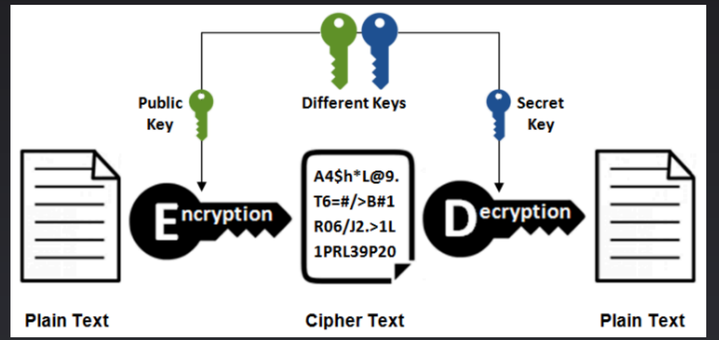
基于密码的身份认证

依托对密码的变换

简单编码(Base64/Base32)
简单哈希(MD5/SHA1/SHA256/...)
加盐、多次哈希

对称密码加密(AES/RC4/...)
非对称密码加密(RSA)





基于密码的身份认证

单纯依赖对密码的变换无法保证用户身份认证的安全性

简单编码(Base64/Base32)
简单哈希(MD5/SHA1/SHA256/...)
多次哈希

对称加密(AES/RC4/...)
非对称加密(RSA)

身份认证

由可预测的数据生成

```
a.c = h.MD5("2989d4f8dcda393d1c1ca3c021f0cb10" + arg2.getPackageName().getBytes());
```

硬编码密钥

```
String v0 = "134e3265829ff82daf16e7b740a600b5";
        if(this.b == null) {
            byte[] v1 = v0.getBytes();
            byte[] v2 = new byte[16];
           ...
            this.b = new SecretKeySpec(v2, "AES");
```

身份认证

RSA/ECB/NoPadding

没有随机性

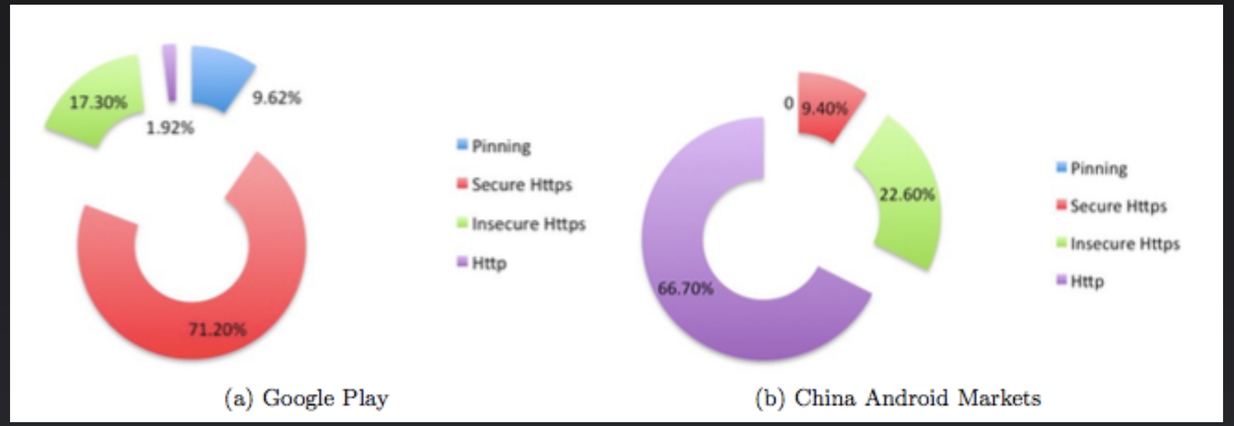
RSA/ECB/PKCS1Padding

除不能抵抗重放攻击外，安全性较高

身份认证

Google Play 100个APP
大陆安卓应用市场200个APP

密码发送通道分布情况



(a) Google Play (b) China Android Markets

身份认证

密码变换种类的使用情况

Type of *process*		China Android Market			Google Play Market		
		HTTP	HTTPS*	HTTPS**	HTTP	HTTPS*	HTTPS**
Trivial Transformation	Plaintext	**44**	**19**	8	**1**	**8**	42
	Encoding	**2**	0	0	0	**1**	0
Hash	One-time MD5	**21**	**8**	5	0	0	0
	Fixed-salt MD5	**6**	0	0	0	0	0
	Multi-time hash	**3**	0	0	0	0	0
Symmetric Encryption	AES/DES with hard-coded key	**24**	**7**	2	0	0	0
	AES/DES with randomly generated key	**1**	0	0	0	0	0
Asymmetric Encryption	RSA/ECB/NoPadding	**3**	**2**	0	0	0	0
	RSA/ECB/PKCS1Padding	2	0	0	0	0	0
	Sum	106	36	15	1	9	42

密钥协商

✖ 双方共享的密钥硬编码在程序中

某通信云提供商的SDK，逆向难度较大，但仍可以提取到密钥

✖ 通信密钥由一方直接发送给另一方

另一即时通信云提供商的SDK，在客户端每次登录成功或者断网重连后服务器会将密钥发给客户端

```
{
  key = 'cd60';
  v84 = 'fa78';
  v85 = '73f5';
  v86 = '400a';
  dstlen = v74;
  v87 = '05ad';
  v88 = 'beec';
  v89 = '1a23';
  v90 = '7c9b';
  v45 = v74 + 9;
  v91 = 0;
  dst1 = operator new[](v74 + 9);
  src_1 = operator new[](v45);
```

攻击者只需获取到网络流量，即可解密获得消息内容

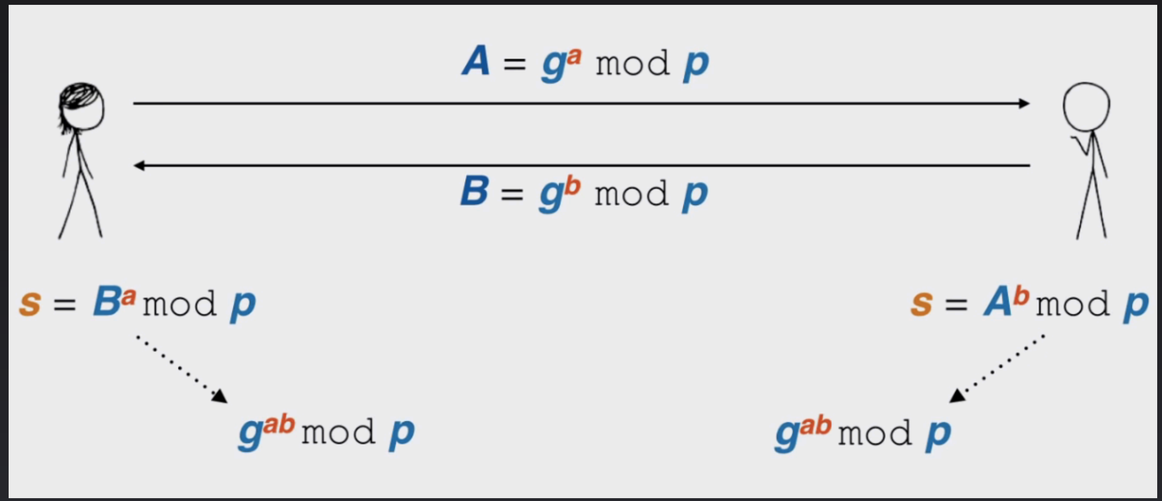
数据包 get √ = 消息裸奔

密钥协商

通信时协商

尽管攻击者可以通过作为中间人分别与双方协商密钥，但避免了攻击者只要获取网络流量就可解密获得消息的情况，增大了攻击难度。

Diffie-Hellman密钥协商



数据加密

考虑到通信效率，采用对称加密算法

避免使用非标准算法

线性分析、差分分析、滑动攻击、...

避免使用ECB模式

CBC模式中IV一定要随机

案例分析

不正确的共享密钥
生成密钥的材料随密文数据一同发送

不需要维护Session
不需要状态切换
任意数据包可解

案例分析

泄露用户隐私

MAC地址、wifi名称、IMEI等

任意篡改、伪造推送消息

点击打开任意网址

点击下载任意app

```
Send '\xe6\xf2\x81\x00\x00\x00\x00\x00\x00\x00\x00\x0817258000\x824{"id":"1457335822066","action":"ad
dphoneinfo","info":{"app":"MDP","model":"SM-N9006","cell":"","imei":"359786050198713","sim":"46001218
2913413","mac":"","type":"ANDROID","system_version":"android4.3","deviceid":"ANDROID-93ee89e0f06d47b8
bd9b34741b71990f","channelid":"open:fm.xiami.main","version":"2.6.0.0"}}\x00 921f7606f5a217900ea4e1e4
6d8e3e2c'
Recv '\xe6\xf2\x81\x00\x00\x01SO\xfb\xf4\\\x0817258000*{"action":"received","id":"1457335822066"}\x00
921f7606f5a217900ea4e1e46d8e3e2c'

Recv '\xefy\x81\x00\x00\x01SP\x00\x85\'\x03OMP\x85^{"id":"ce5699d8-5457-4374-8e3d-71aa0c08c44e_OM
P","condition":{"duration":""},"taskid":"OSA-0307_d1yBLcjHwb5HT3mR6idYl7","action_chains":[{"actioni
d":"1","do":"10030","type":"goto","is_withnettype":"false"},{"actionid":"10030","noinstall_action":"1
00","appid":"","do":"100","appstartupid":{"android":"","ios":"","symbia":""},"type":"startapp","is_wi
thnettype":"false","is_autostart":"false"},{"actionid":"100","type":"null","is_withnettype":"fals
e"}],"appkey":"jfY3OTRHNC9tSAvPTxK3F2","action":"pushmessage","appid":"UgM0PuGY4Z9hhSeWysmxz5","messa
geid":"ce5699d8-5457-4374-8e3d-71aa0c08c44e","push_info":{"message":"","action_key":"","launch-imag
e":"","sound":"","loc-args":"","payload":"","action-loc-key":"","loc-key":"","badge":""}};{"title":"华
晨宇新歌《横冲直撞》独家首发!","text":"邀你一起回忆我们滚烫炙热的青春!","exts":{"category":"system","category_i
d":86,"name":"虾小米","url":"xiami:\/\/open?url=http%3a%2f%2fwww.xiami.com%2fmarket%2fmusic%2fact%2fhu
achenyu.php","scheme_url":"xiami:\/\/open?url=http%3a%2f%2fwww.xiami.com%2fmarket%2fmusic%2fact%2fhua
chenyu.php","image":"http:\/\/img.xiami.net\/images\/common\/uploadpic\/68\/14573234681214.jpg","typ
e":"artist"}} 921f7606f5a217900ea4e1e46d8e3e2c'
Send 'x"\x81\x00\x00\x00\x00\x00\x00\x00\x00\x0817258000\x82\x14{"action":"pushmessage_feedback","app
id":"UgM0PuGY4Z9hhSeWysmxz5", "id":"1457336121378", "appkey":"jfY3OTRHNC9tSAvPTxK3F2", "messageid":"c
e5699d8-5457-4374-8e3d-71aa0c08c44e","taskid":"OSA-0307_d1yBLcjHwb5HT3mR6idYl7","actionid": "0","resu
```

认证消息构建

认证消息

保证消息不被篡改

使用简单哈希构建签名

HMAC密钥泄露

结合密钥交换或数据加密模式

一些讨论

在安全的信道中传输广义意义上的私有协议数据是安全的做法
密码学安全前提不满足是造成实际中密码学漏洞的本质原因
密码学安全问题造成的后果视使用场景而定
Security by Obscurity是不可取的

THANKS

乌云 WooYun

乌云白帽大会 · 2016
不插电