第三讲 身份认证技术

测试点 3-1

1、编制一个口令强度检测程序（语言不限）要求如下：

口令必须包含大写字母，小写字母，数字，特殊字符四种中的三种，长度要求 8 到 30 位。

源代码

```
import java.util.Scanner;

public class checkPasswd {



    public static void main(String[] args){
       Scanner in=new Scanner(System.in);

       for(;;){
           System.out.print("请输入口令:");
           String password=in.next();

           int upperNum=0,lowerNum=0,intNum=0,otherNum=0;
           for(int num=0;num<password.length();num++){

    if((int)password.charAt(num)>64&&(int)password.charAt(num)<91)
upperNum++;
               else
if((int)password.charAt(num)>96&&(int)password.charAt(num)<123)
lowerNum++;
               else
if((int)password.charAt(num)>47&&(int)password.charAt(num)<58)
intNum++;
               else otherNum++;
           }
           if(password.length()<8||password.length()>30){
               System.out.println("密码长度必须大于8位小于30位\n请重新
输入密码");
               continue;
           }
           else
if(upperNum*lowerNum*intNum==0&&upperNum*lowerNum*otherNum==0){
               System.out.println("密码必须包含大小写、数字、特殊字符中
```

```
的三种及以上\n请重新输入");
                continue;
            }
            else{
                System.out.println("密码符合要求");
                break;
            }
        }
        in.close();
    }

}
```

运行结果

```
<terminated> checkPasswd [Java Application] D:\myeclipse\binary\
请输入口令:hkoi67
密码长度必须大于8位小于30位
请重新输入密码
请输入口令:897897ghg
密码必须包含大小写、数字、特殊字符中的三种以上
请重新输入
请输入口令:JT67889nj
密码符合要求
```

测试点 3-2

1、针对基于密码的身份认证，主要存在重放攻击和中间人攻击两种安全风险，试分析基于对称密码的 Needham-Schroeder 协议流程，判断该协议是否存在安全风险？如果存在，请给出攻击过程和改进方案。

这个协议的一个攻击是 Denning 和 Sacco 在 1981 年发现的，其主要问题是主体 B 无法 确定消息 M3 是否是新鲜的。一个攻击可以破解一个密钥之后替换发给主体 B 的消息 M3 并 完成协议。同样，主体 A 也可以对消息 M3 进行处理，比如发送一个旧的 Kab 给主体 B。为 了解决这个新鲜性问题，Denning 和 Sacco 提出了用时戳的方法和引入公钥证书的方法，并 提出了 Denning-Sacco 协议，但该协议同样有问题，可以参考文献[9]。另外，无需获得泄漏的会话密钥也可以实施一种攻击，具体描述如下：

$$M1 \quad A \rightarrow S: A, B, N_a$$

$$M2 \quad S \rightarrow A: \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$$

$$M3 \quad A \rightarrow Z(B): \{K_{ab}, A\}_{K_{bs}}$$

$$M4 \quad Z(B) \rightarrow A: N_z$$

$$M5 \quad A \rightarrow Z(B): \{\{N_z\}_{K_{ab}} - 1\}_{K_{ab}}$$

攻击者 Z 截获了 A 发送给 B 的消息，并伪装成 B，给 A 发送一个与{ Nb} K ab 格式相同 的随机数 Nz ，于是 A 依协议要求对 Nz 进行解密，实际上是对 Nz 进行了加密，将所得结果 { Nz } K ab 减 1 并加密后发回给 B，Z 也截获了此消息。这时，A 认为主体 B 已经知道了会话密钥 Kab ，但 B 实际上并没有参与协议的执行过程。

针对上述 Needham-Schroeder 共享密钥协议的缺陷及容易受到的攻击，我们将协议修改为：

$$M1 \quad A \rightarrow S: A, B, N_a$$

$$M2 \quad S \rightarrow B: A$$

$$M3 \quad B \rightarrow S: N_b^0$$

$$M4 \quad S \rightarrow A: \{N_a, B, K_{ab}\}_{K_{as}}$$

$$M5 \quad S \rightarrow B: \{K_{ab}, A, N_b^0\}_{K_{bs}}$$

$$M6 \quad B \rightarrow A: \{N_b, K_{ab}\}_{K_{ab}}$$

$$M7 \quad A \rightarrow B: \{K_{ab}, N_b - 1\}_{K_{ab}}$$

2、使用公钥密码的 Needham-Schroeder 协议流程如下：

1. $A \rightarrow B: E_B(N_A, A)$
2. $B \rightarrow A: E_A(N_A, N_B)$
3. $A \rightarrow B: E_B(N_B)$

分析该协议中存在的安全风险，给出攻击过程和改进后的方案。

该协议的缺陷在于无法验证发消息的是确信的实体，易受到第三方的重放攻击。改进方案是在第二、三步加入相应的签名。