<EditText

android:id="@+id/etPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:hint="请输入密码"

android:inputType="textPassword"

android:textColor="#000000"

android:textColorHint="#55000000"

android:textSize="20sp"/>

<**EditText  
 android:layout\_width="match\_parent"  
 android:layout\_height="45dp"  
 android:textSize="19dp"  
 android:gravity="left|center\_vertical"  
 android:background="@null"  
 android:id="@+id/pwd\_a"  
 android:textColorHint="@color/textColorHint"  
 android:textColor="@color/account"  
 android:inputType="text"  
 android:paddingLeft="3dip"  
 android:password="true"  
 android:hint="@string/enter\_password"**/>

*方式*1mEtPassword.setInputType(InputType.***TYPE\_TEXT\_VARIATION\_VISIBLE\_PASSWORD***);  
*//默认状态显示密码--设置文本 要一起写才能起作用 InputType.TYPE\_CLASS\_TEXT | InputType.TYPE\_TEXT\_VARIATION\_PASSWORD*mEtPassword.setInputType(InputType.***TYPE\_CLASS\_TEXT*** | InputType.***TYPE\_TEXT\_VARIATION\_PASSWORD***);

*方式*2 和方式1的不同, 方式1会显示一下原始输入，然后再转成系统默认替换符

mEtPassword.setTransformationMethod(HideReturnsTransformationMethod.getInstance());

mEtPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());

既可以显示一下原始输入，然后再转成系统默认替换符，又可以设置系统替换符，自定义，

自定义，

监听粘贴，大于一个字符，立刻显示替换符

输入一个字符，显示一会（1s），然后转成替换符

输入一个字符，把之前的都设置成替换符，清除对应的\*handler

也可以用这个种SpannableString 来做，更灵活，方便，可以显示图片

//自定义

mEtPassword.setTransformationMethod(AsteriskPasswordTransformationMethod.getInstance());

**public class** AsteriskPasswordTransformationMethod **extends** PasswordTransformationMethod {  
   
 @Override   
 **public** CharSequence getTransformation(CharSequence source, View view) {   
 **return new** PasswordCharSequence(source);   
 }   
   
 **private class** PasswordCharSequence **implements** CharSequence {   
 **private** CharSequence **mSource**;   
 **public** PasswordCharSequence(CharSequence source) {   
 **mSource** = source; *// Store char sequence* }   
 **public char** charAt(**int** index) {   
 **return '\*'**; *// This is the important part* }   
 **public int** length() {   
 **return mSource**.length(); *// Return default* }   
 **public** CharSequence subSequence(**int** start, **int** end) {   
 **return mSource**.subSequence(start, end); *// Return default* }   
 }