利用声音掩蔽技术保护智能手机中的音频隐私

董玉芝
密歇根大学 yctung@umich.edu

Kang G. Shin
密歇根大学
kgshin@umich.edu

摘要

通过恶意软件未经授权的录音泄露隐私已成为移动用户面临的新威胁。为了应对这一威胁，我们提出了 SafeChat，它可以防止未经授权的录音，而无需在操作系统中进行新的音频隐私设置。SafeChat 利用声音屏蔽技术来区分授权和非授权录音应用程序之间的音频信息。具体来说，即使授权录音应用程序和未授权应用程序从同一个麦克风中录制了相同的音频信号，SafeChat 也能使授权录音应用程序比未授权应用程序恢复更多的优先/机密信息。我们将 SafeChat 作为一款安卓聊天应用来实现。我们在几款商品手机上进行的实验表明，SafeChat 可以使授权和未授权录音应用程序之间的信号强度相差高达 26dB。这种差异会降低谷歌语音应用程序接口（Google Speech API）等最先进语音识别引擎的准确率，使其对未经授权录音的理解率低于 0.1%，而对授权录音的理解率却很高。此外，在我们在线招募的 317 名测试参与者中，没有一人能理解屏蔽语音。我们的可用性研究表明，只有 35% 的参与者意识到了隐私泄露的威胁，60% 的参与者希望使用安全聊天工具来保护他们的私人/秘密信息免遭未经授权的录音。

中央案例研究的概念

• **安全和隐私 → 移动平台安全**；*隐私保护协议*； - **网络 → 移动和无线安全**。

关键词

音频隐私；声音屏蔽；移动系统

ACM 参考格式：

Yu-Chih Tung and Kang G. Shin.2019.利用声音掩蔽实现智能手机中的音频隐私。In *ACM Asia Conference on Computer and Com-munications Security (AsiaCCS '19), July 9-12, 2019, Auckland, New Zealand.*ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3321705.3329799

亚洲计算机会议'19，2019 年 7 月 9-12 日，新西兰奥克兰

https://doi.org/10.1145/3321705.3329799

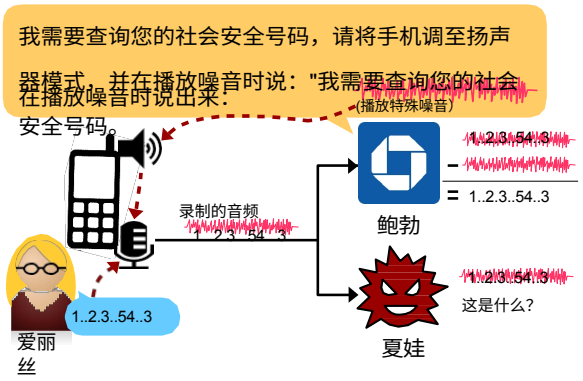


图 1：私人/机密信息受声音屏蔽保护。安装在用户手机中的音频恶意软件无法理解嗅探到的秘密，因为该信息被内置接收器产生和添加的噪音所混淆。

1 引言

通过手机麦克风泄露隐私信息已成为移动用户的一大担忧。在后台偷偷记录对话的间谍软件在应用市场上越来越受欢迎[4, 6, 8]。还有研究表明，恶意软件可以通过最先进的语音识别引擎轻松识别用户的信用卡号码 [24]。如果我们考虑到在许多应用程序中嵌入相同的利用程序的可能性，而这些应用程序已经从用户那里获得了音频访问权限，那么这种威胁就会变得更加令人担忧。为了应对这种威胁，我们提出了 SafeChat，这是一种防止通过手机麦克风*进行未经授权的录音*而泄露隐私信息的新方法。具体来说，SafeChat 主要针对图 1 所示的常见威胁场景，即用户（Alice）通过电话向服务提供商（Bob）告知自己的秘密信息，而恶意软件（Eve）正在录音，然后偷偷泄露录音信息。

解决这一问题的传统方法是向应用程序提供虚假音频数据，或根据细粒度传感器隐私策略阻止音频访问 [13, 16, 17, 22, 29]。然而，这些解决方案都需要对手机权限控制系统进行重大修改，而且用户要选择*正确的*权限/隐私策略也并非易事。此外，服务提供商（通常是请求秘密信息的一方）仍然无法判断用户是否在预期的隐私策略设置下启用了此类安全功能。例如，服务提供商无法确定用户手机上哪些应用程序可以访问设备的麦克风，也无法确定所有这些应用程序是否都值得信赖。

我们的解决方案 SafeChat 可防止未经授权的录音泄露隐私，而无需任何额外的音频权限控制，并让用户和服务提供商都能

提供商在必要时调用这一安全功能。SafeChat 利用*声音屏蔽*来实现上述保护。图 1 显示了一个银行代表 Bob 通过电话向客户 Alice 索取私人信息的例子。在这种情况下，鲍勃首先要求爱丽丝将手机调到扬声器模式，然后一边播放遮蔽声音/噪音一边说出所要求的信息（如后面所述，SafeChat 可以自动处理这一操作）。手机麦克风将记录 Bob 播放的噪音和 Alice 说出的私人信息。这样，私人信息就会被这种叠加噪音所掩盖，只有鲍勃才能恢复。未经授权的应用程序无法移除掩蔽声音，因为只有鲍勃知道这种噪音是如何产生和添加的。

SafeChat 的灵感来源于：(i) 现有解决方案需要额外的用户自定义策略；(ii) 授权和未授权的录音应用程序都会收到相同的录音副本，而且应用程序中没有明确区分秘密对话和正常对话。SafeChat 通过将隐私保护从移动设备转移到服务提供商（以下称为*目标接收方*）来解决这一问题。这种设计有两个明显的优势。首先，隐私保护可以远程计算，从而使 SafeChat 向后兼容（有关支持的设备和系统要求，请参阅第 5 章和第 7 章）。

安全聊天）。其次，预定接收方可以在必要时启动安全聊天频道[1]。例如，用户仍然可以使用通话录音应用程序保存大部分对话内容。但是

只要接收方要求用户提供私人/秘密信息，就可以启用隐私保护，只有这部分对话才会被 SafeChat 从通话录音应用程序中混淆。

要实现 SafeChat，需要克服几个挑战。其中之一是，由于音频传输的多径和共振特性以及设备扬声器/麦克风的失真，麦克风记录的掩蔽声音与原始音频信号并不完全相同。因此，有必要设计一种适当的声音掩蔽（噪音）信号及其移除算法，既能有效地混淆私人信息，又能使预期接收者移除掩蔽声音。为了验证 SafeChat 的功能，我们将其作为一款安卓聊天应用来实施。我们假定通信是通过我们的应用程序建立的，并且当按下应用程序上的特定（软）按钮时，安全屏蔽声音可以被播放/移除。以其他形式实现 SafeChat 是我们未来工作的一部分，例如作为现有聊天应用程序的第三方库，或作为支持普通电信的系统服务。

我们在多款安卓设备上的实验结果表明，在未经授权的录音中，增加的掩蔽噪声可比在预定接收器上恢复的录音高出 26 分贝。假设恶意软件可以通过语音识别引擎或众包[24]轻松识别所记录的秘密，那么当使用最先进的谷歌语音应用程序接口[5]进行识别时，SafeChat 可以将 8 位数嗅探秘密的单词准确率从 99% 降低到 0.1%。我们的信号移除算法可以将识别准确率恢复到 95%，从而产生显著的差异

[1] 双音多频（DTMF）信令是通过拨号音发送私人信息的另一种方式，但它也容易受到未经授权的录音的影响[26]。

在这种情况下，DTMF 信号可能会被目标接收者和未经授权的录音应用所理解。此外，当要求重新在线招募的 317 名参与者 "识别隐藏在噪音中的数字 "时，他们中没有人能完全识别出被掩盖的秘密/数字，而预期接收者却能正确地恢复它们。第 4 章和第 7 章还讨论了攻击者为消除这种掩蔽声音而可能进行的预处理/过滤的影响。

本文有以下 4 个主要贡献：

- 在安卓设备上设计应用程序级保护，以提供基于声音掩蔽的音频隐私[第 3 节]；
- 对移动设备上的声音掩蔽进行安全分析，以防止未经授权的录音[第 4 节]；
- 广泛的评估表明，谷歌语音应用程序接口和 317 位测试参与者对授权录音的理解准确率高于 95%，而对未经授权录音的理解准确率低于 95%。

  而未经授权的录音则无法理解[第 5 节]；以及
- 衡量安全聊天软件的可用性并展示其在现实世界中的优势 [第 6 节]。

2 相关工作

通过手机麦克风泄露隐私（即*未经授权的录音*）已成为智能手机用户面临的一个新威胁，因为任何已安装的具有音频访问权限的应用程序都可以在任何时间偷偷录制任何信息。利用这种泄漏的恶意软件可以轻易收集用户的信用卡信息和社会安全号码 [24]。针对这种未经授权的录音行为，现有的防御措施通常是向应用程序提供虚假音频数据，或根据预先定义的传感器隐私策略禁用录音功能。例如，一些系统[17, 29]建议在电信开启时始终向应用程序提供伪造的音频数据，而另一些系统[22, 31]则创建了一个类似格子的隐私保护层来表示应用程序之间的冲突，并在适当的时候禁用音频记录功能。文献[16]提出了一种可编程的传感器隐私策略，这样受信任的第三方就可以为移动用户设计适当的策略来控制录音。然而，一般来说，很难定义 "适当的 " 隐私政策。例如，怎样才能在启用电话录音应用程序的同时防止私人信息泄露给同一应用程序？此外，所有这些解决方案都需要对现有的隐私控制系统进行重大修改，因此在商品移动设备中部署的可能性较小。

SafeChat 是一种应用级解决方案，它通过声音掩码保护音频隐私，无需对现有系统进行任何修改。*声音掩蔽*是一种音频混淆技术，用于防止通过窃听或附近设备的窃听泄露隐私[28]。传统声音掩蔽的设计假设是，窃听设备记录的音频质量或信号强度较低（由于距离窃听设备较远或窃听设备失真），因此会在背景中产生足够的噪音，以防止窃听者理解口语信息，同时保留预期接收者理解信息的能力。最近，有人利用声音掩蔽技术，通过安装在天花板上的扬声器阵列产生的波束成形噪声，建立了一个没有墙壁但隐私得到保护的办公室[3, 7]。然而，这种使用案例并不适用于移动场景，因为安装的恶意软件和

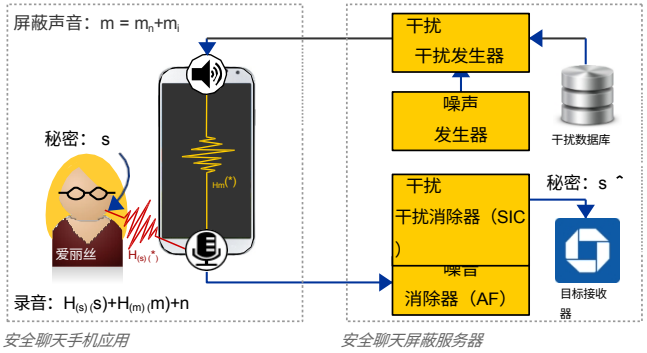


图 2：系统概述。掩蔽声音 m 和秘密信息 s 通过不同的路径 $H_{(s)}(*)$ 和 $H_{(m)}(*)$ 发送到麦克风，因此需要进行特殊的信号处理，才能在目标接收器处从录制的音频中去除掩蔽声音。

在目标接收器处，从录制的音频中去除掩蔽声音。

目标接收方从同一个手机麦克风录制的信号完全相同，因此添加噪音会混淆两者。SafeChat 解决这个问题的方法是先生成特殊的掩蔽声音，然后再在预定接收方处将其移除，从而实现应用级保护，这就相当于在未授权应用和授权应用之间提供不同的音频隐私保护。

与 SafeChat 最接近的是 mSieve [23]，它依靠不同的隐私向某些应用程序提供比其他应用程序更多的音频信息。但是，它没有解决如何处理信号以推导数学模型的问题，而且还需要修改系统，以便在应用程序访问音频信号之前对其进行预处理。无线网络中的非加密安全解决方案也与 SafeChat 有关。例如，发射器可以在信道状态信息（CSI）的空域处向目标接收器生成人工噪声[11, 14]，这样，位于与目标接收器不同位置的窃听者就无法移除添加的噪声并正确解码嗅探数据包。目标接收器也可以在接收数据包时广播人工噪声，并通过基于 CSI 的信号去除过程恢复秘密 [27]。尽管这些无线安全解决方案的概念与 SafeChat 相似，但它们无法直接解决未经授权的录音问题。例如，人的声音并不像无线信号那样被调制/编码，因此没有 OFDM 前导码可以用来估计 CSI。与此相反，SafeChat 采用了一种新颖的方法，即通过自适应滤波估计从扬声器到麦克风的信道响应，然后通过连续干扰消除（SIC）去除残留的附加噪声 [25]。其他研究人员也探索了利用声音信号建立安全数字信道的方法，例如 Dhwani [20] 和 PriWhisper [30]。SafeChat 与这些系统有相似之处，但针对的是不同的问题：项目敏感转换，而不是在声音中编码秘密信息。

3 系统设计

图 2 展示了 SafeChat 的概览。当启用它来传输秘密信息 s 时，设备的扬声器会播放掩蔽声音 m。

由设备的扬声器播放。由于声音是通过多条路径在空中传播的，因此录制的声音实际上是由多个延迟和衰减的原始播放声音的副本组合而成。的多个延迟和衰减副本的组合。假设这种组合是线性的，那么可以用通道响应 $H(*)$来表示声音在麦克风处如何组合以进行录音，即 $recorded(sound) = H(*)$。

H（$played(sound)$）。我们将用 $H_{(s)(*)}$ 来表示信道重

我们将用 $H_{(s)(*)}$ 表示爱丽丝说出的秘密信息的信道响应，而用 $H_{(m)(*)}$ 表示播放的掩蔽声音的信道响应。因此，记录的音频 r 变成

$$recorded(sound) = r = H_{(s)(s)} + H_{(m)(m)} + n、 \quad (1)$$

其中 n 为高斯环境噪声。SafeChat 能否有效消除未经授权的录音取决于屏蔽声音的选择。例如，将掩蔽声音作为一段音频发送，其中包含数百万个来自不同人的口语句子，应该能够有效地隐藏秘密，因为听者无法分辨哪句话是秘密。但是，这种设计对预期接收者来说也会有问题，因为预期接收者不知道录音中的掩蔽声音有多 "失真"，也不知道如何消除失真源。此外，如果攻击者知道用户声音的适当上下文（如音调），那么聊天中的秘密仍有可能被专门设计的过滤器提取出来。因此，设计适当的掩蔽声音并确保在目标接收器中去除该声音对安全聊天功能的有效运行至关重要。

3.1 选择掩蔽声音

安全聊天软件将掩蔽声音选择为两个信号成分的组合，即 $m = m_i + m_{(n)}$。第一部分是掩蔽干扰，即 m_i，其中包括几个预先录制的人说句子，以混淆和防止恶意软件提取秘密信息。由于我们的主要目标是保护信用卡或社保号码等秘密信息，因此我们从一个名为 TIDIGITS [18] 的声音数据库中选择了这种干扰，该数据库包括数字的音频片段。请注意，为了进一步提高系统性能，我们还可以从其他来源（也可以由 Alice 录制）选择这种屏蔽干扰，这也是我们未来工作的一部分。

第二个成分是掩蔽噪声 m_n，它是以高斯噪声的形式产生的，并由 16kHz 低通滤波器滤除。SafeChat 只保留 16kHz 以下的噪声频率，因为该频率范围涵盖了大部分人类语音频谱。添加这种高斯噪声有助于降低秘密信息的信噪比，这相当于恶意软件录制音频的语音智能[10]。此外，它还有助于目标接收器恢复秘密信息，避免基于滤波器的掩蔽声音分离，例如，只保留用户音调范围内的声音。

需要注意的是，在我们的实施过程中，先导信号由几个 10k-24kHz 的啾啾声组成，在掩蔽声音之前播放。这个先导信号的目的是帮助同步设备扬声器和麦克风之间的时间偏移，因为从程序要求播放声音到实际播放声音之间会有几百毫秒的延迟（这是由于商品手机的非实时操作系统造成的）。如果没有这种同步，就需要在自适应滤波器中设置更大的深度，以确定信道响应的特征，这就需要多花费 100 倍的计算时间。

3.2 去除掩蔽噪声和干扰

有效消除目标接收器的噪声和干扰对模糊机密信息至关重要。与无线系统中广泛使用的基于 CSI 的技术移除附加噪声不同 [14, 27]，移除智能手机中的掩蔽声音并非易事。

与无线系统中广泛使用的基于 CSI 的技术[14, 27]不同，由于对信道响应 $H_{s)(*}$和 $H_{(m)(*)}$ 的不了解，要消除智能手机中的掩蔽声音并非易事。

为了解决这个问题，安全聊天使用掩蔽噪声成分来估计信道响应 $H_{(m)(*)}$。具体来说，我们将掩蔽干扰视为秘密信息的一部分，并使用现有的自适应滤波器

现有的自适应滤波器 [15] 将掩蔽噪声从信道响应 H（m）中分离出来。

其他信号。在我们目前的设置中，这种自适应滤波器的深度为

深度设置为 500 个采样点，以处理不同秒钟的声音延迟传播。

设备。这一过程可以通过最小化以下值来确定信道响应的最佳估计值 $\hat{H}_{(m)(*)}$

$$e(\hat{H}_{(m)(*)}) = r - \hat{H}_{(m)(mn)}、 \quad (2)$$

其中，$e(\hat{H}_{(m)(*)}$表示自适应滤波的残余误差，假设添加的噪声与人声。需要注意的是，不可能分离出

由于添加的人声与秘密信息（也是人声）之间的相关性不为零，因此可以使用相同的自适应滤波器滤除掩蔽干扰成分。

根据自适应滤波器的理论，残余er-

或 $e(\hat{H}_{m)(*)}$表示秘密信息 s 的组合、

和添加的屏蔽干扰 m_i。然后，安全聊天应用

通过连续干扰消除（SIC），从残余噪声中消除 $\hat{H}_{(m)(mi)}$ 来恢复秘密信息：

$$\hat{s} = e(\hat{H}_{(m)(*)}) - \hat{H}_{(m)(m(i))}、 \quad (3)$$

其中，$\hat{s}$ 是目标接收器恢复的秘密。注

窃听应用程序无法使用相同的处理方法来提取秘密信息，因为掩蔽声音是由目标接收器生成的，而且只有目标接收器知道。图 3 显示了 Nexus 6P 记录的屏蔽音频的噪声和干扰去除过程示例。在这个例子中，从录制的音频中去除了超过 15 分贝的掩蔽噪声和干扰，从而在授权和非授权录制之间产生了足够的录制信息差距。

3.3 声音屏蔽指标

图 4 显示了安全聊天屏蔽/恢复 3 位数秘密的能量包络示例，以及 4 个重要声音屏蔽指标的定义。此后，我们将使用掩蔽声音与噪声之比（MNR）和掩蔽声音与残余噪声之比（MRR）分别表示添加的掩蔽声音的强度和去除该掩蔽声音的效果。掩蔽声与噪声比

前者（后者）定义为 $r\|\|n\|(\|s\|\|\hat{s}_{nonseech}\|)$，其中噪声项 n *代表背景噪声（而非掩蔽噪声）*。

噪声项 n 代表背景噪声（而非掩蔽噪声

而 $\hat{s}_{nonspeech}$表示恢复信号中没有用户语音的残余噪声。

另一方面，掩蔽声音与语音比（MSR）和语音与恢复噪音比（SRR）分别表示从恶意软件中隐藏的秘密信息量和在目标接收器中恢

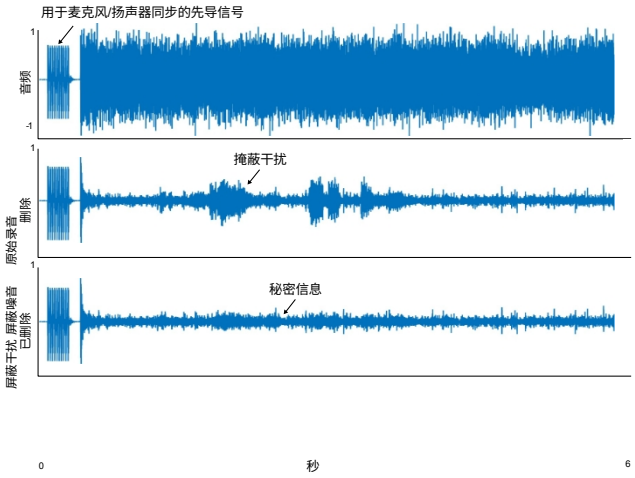


图 3：去除掩蔽声音的示例。首先通过自适应滤波器去除掩蔽噪声，然后通过连续干扰滤波器去除掩蔽干扰。

$\|m\|/\|\hat{s}_{speech}\|$，而后者的定义为 $\|\hat{s}_{speech}\|/\|\hat{s}_{nonspeech}\|$。

语音/非语音信号的部分可以通过标准的语音活动检测（VAD）算法（如 G.729 [12]）来识别。为确保 SafeChat 的处理延迟较低，我们只实施了基于能量阈值的简单 VAD。具体来说，我们使用 $\hat{s}$ 能量包络的 80% 和 20% 百分位数来表示

$\|\hat{s}_{speech}\|$和 $\|\hat{s}_{nonspeech}\|$。如图 4 所示

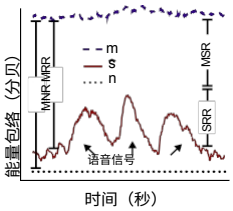
复的秘密信息量。前者定义为

这种简化使得 $s\hat{}_{语音}$被口语秘密中最响亮的部分所支配，反映了 SafeChat 防止整个口语秘密泄漏的目的。在没有语音信息记录的少数情况下，这种简化可能会略微高估 MRR。这个问题不会对我们的评估造成太大影响，因为大多数测量结果都包含了口语秘密，而且安全聊天的最终性能是以屏蔽秘密和恢复秘密之间的识别准确率差异为特征的。因此，在不失一般性的前提下，为了使阈值设置保持一致，我们选择在此基础上报告 MRR。

总之，MNR 和 MRR 描述了设备的硬件即与 $s\hat{}_{语音}$无关，而 MSR 和 SRR 则反映了安全聊天软件在防御未经授权的录音方面的性能。理想情况下，当这四个指标的值尽可能高时，安全聊天软件就能达到最佳性能。然而，这些指标是相互关联的。例如，高 SRR 意味着低 MSR，因为去除的掩蔽声音是有限的。接下来将详细介绍我们在这些指标之间寻求平衡的设计选择。

3.4 设备音量控制

语音智能与语音的信噪比（SNR）相对应[10]，而信噪比与掩蔽声音的音量成负相关。因此，掩蔽声音的播放音量应足够大（即高信噪比），以降低秘密信息的语音智能。但是，由于硬件的限制，掩蔽声音的音量不能无限大。此外，播放过大的掩蔽声音也是不明智的，因为线性信道响应的假定会失效。



指标	描述
MNR	掩蔽声音与噪声的比率
MRR	掩蔽声与残余噪声比
MSR	掩蔽声与语音信号比
SRR	语音信号与残余噪声比

图 4：声音掩蔽指标解释。MNR 和 MRR 表示设备播放/移除遮蔽声音的能力，而 MSR 和 SRR 则表示有多少声音被遮蔽。向恶意软件和内置接收器泄露/接收秘密信息。

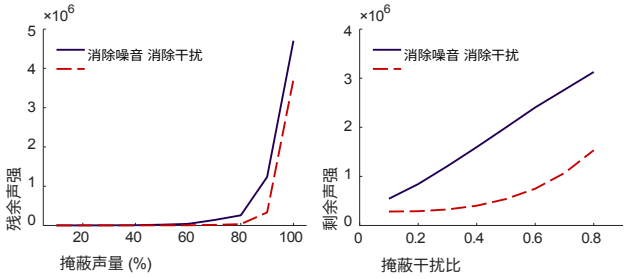


图 5：声音掩蔽的效果。音量越大，对声道响应的估计就越准确。然而，当用最大音量播放掩蔽声音时，声道响应会变得非线性，从而留下较大的残余误差。

当播放/录制的声音摇摆到手机扬声器/麦克风的非线性区域时。另一方面，用小音量播放掩蔽声音不仅无法隐藏秘密信息，而且由于对信道响应的估计不准确，在我们去除掩蔽声音的过程中会留下更多的残余噪声。

图 5(a)显示了在 Nexus 6P 上以不同扬声器音量播放遮蔽声音并去除遮蔽声音后的平均声音强度。在这个例子中，音频是在安静的环境中录制的，不包括任何有声语言，因此图 5(a) 中显示的残余声强代表了我们的噪声和干扰去除算法的残余误差。如图所示，当播放的声音超过最大音量的 90% 时，自适应滤波器的残余误差会因非线性失真而激增，从而阻碍了在这种高音量掩蔽声音下的恢复过程。

除了选择最佳扬声器音量来播放整个掩蔽声音外，我们还需要找到平衡点，将有限的音量预算用于掩蔽噪声和干扰。如果播放的掩蔽干扰音量小于记录的秘密信息的音量，则毫无意义。但是，如果使用过高的音量来播放掩蔽干扰，就会增加用 SIC 消除干扰的残余误差，从而使目标接收器难以理解恢复的音频。这是用 SIC 消除信号的常见问题，当先消除的信号（即掩蔽噪声）的信号强度大于后消除的信号（即掩蔽干扰）时，系统就能达到最佳性能[25]。图 5(b) 显示了将设备扬声器音量固定在最大音量的 90%，同时改变音量比例来播放掩蔽干扰的例子。如图所示，播放掩蔽干扰的音量比越大，不仅自适应滤波的残余能量越多，而且由于信道响应估计不准确，SIC 的残余误差也会增大。在我们目前的设计中，无论扬声器音量有多大，掩蔽干扰的能量比始终固定为比掩蔽噪声音量低 10dB 。这一选择是基于我们的实验结果，以及秘密语音的能量比播放的掩蔽声音低 13 分贝的假设。第 5 节将详细介绍这种掩蔽干扰设置的性能。

算法 1 设备校准

输入： 录制的音频：$r(vol, ch)$ 和 MRR 门限：thr_{MRR} **输出：** 麦克风声道和扬声器音量：ch_{calib}, $vol_{(calib)}$ 1: $n = estimate\_background\_noise$ (r)
2: $s\hat{}(vol, ch) = 移除\_masking\_sound\ (r\ (vol, ch))$
3: $MNR(vol, ch) = 20\log10(\|r(vol, ch)\| / \|n\|)$
4: $MRR(vol, ch) = 20\log10(\|r(vol, ch)\| / \|s\hat{}_{(nonspeech)}(vol, ch)\|)$
5: 最小 $MNR = inf$，最大 $MRR = -inf$
6: **for** ch for all microphone channels **do** 7:
$meanMNR = mean(MNR(:, ch))$ 8: **if**
$minMNR > meanMNR$ **then**
9: $ch_{calib} = ch$
10: $minMNR = meanMNR$
11: **for** vol for all volumes **do**
12: **if** $maxMRR < MRR(vol, ch_{calib})$ **then**
13: $vol_{calib} = vol$
14: $maxMRR = MRR(vol, ch_{calib})$
15: **如果**最大 $_{MRR} < thr_{MRR}$ **则**
16: **返回**失败
17: **对于**从最大容量到最小容量的 vol，**执行**
18: **if** $|maxMRR - MRR(vol, ch_{calib})| < 1dB$ **then**
19: $vol_{calib} = vol$
20: $break$
21: **返回** ch_{calib} & vol_{calib}

3.5 设备校准

如前所述，SafeChat 需要找到合理的音量来播放掩蔽声音，并找到适当的音量比例来播放掩蔽干扰。同时，还需要找到一个合适的麦克风作为参考来恢复秘密信息。在我们的实验中，由于麦克风的位置不同，一个麦克风记录的掩蔽声音信号强度可能比另一个麦克风高 20dB。因此，如算法 1 所示，SafeChat 会进行一次性校准，以寻找最佳麦克风作为参考，并寻找最佳扬声器音量来播放掩蔽声音。

在一次性校准过程中，安全聊天软件会自动播放不同扬声器音量（vol）的掩蔽声音，并通过不同设备麦克风（ch）进行录音。

不同音量设置的录音信号 $r(vol, ch)$ 首先由我们的屏蔽声音移除算法进行处理，以恢复秘密信息 $s\hat{}(vol,ch)$。然后，我们估算所有音量和信道设置的 MNR 和 MRR。在校准过程中，参考音量（vol, ch 用于恢复秘密的手机是根据 MNR 选择的。具体来说，安全聊天总是选择 MNR 最低的麦克风作为参考，因为低 MNR 意味着麦克风接收到的掩蔽声音较少（由于位置离说话者较远或麦克风增益设置不同）。SafeChat 有必要通过接收足够强的掩蔽声音来确保保护这个 "最弱 "的麦克风。在我们的实验中，当掩蔽声音的音量调整到足以保护这个参考麦克风时，它自然会使移动设备上的另一个麦克风达到饱和。

一旦确定了参考麦克风的麦克风通道 ch_{clib}，SafeChat 就会寻找 MRR 最高的音量，该音量代表了 SafeChat 消除掩蔽声音的能力。声音有效。SafeChat 为通过最大 MRR 设置了一个阈值。根据我们目前对设备的测试，该阈值被设置为 18dB。如果无法找到满足此条件的任何掩蔽音量，则校准失败。请注意，基于最高 MRR 的音量选择是为了确保恢复秘密。然而，安全聊天软件的主要目标仍然是防止未经授权的录音，因此我们的校准过程会积极搜索具有类似 MRR 的（即比最大 MRR 小 1dB）的更高音量（如果有的话），并将该音量设置为校准音量 vol_{calib}。不同设备的校准结果将在第 5 节中介绍。

3.6 语音音量检查

除了设备播放和移除掩蔽声音的能力外，用户说话的音量对系统性能也至关重要。如果秘密说得太大声，所选的掩蔽声音可能无法有效地混淆秘密。另一方面，说话音量太小可能会使目标接收者无法理解恢复的秘密。

解决这个问题的一个可行方法是根据用户说话的音量动态调整掩蔽声音的音量。不过，如上所述，掩蔽声音不应太大，以免进入麦克风/扬声器硬件的非线性范围。此外，根据我们的初步实验结果，当提高掩蔽声音的音量时，用户往往会提高自己的音量，从而抵消了动态调整掩蔽声音音量的效果。目前，SafeChat 通过一次性设备校准来固定掩蔽音量，并在说出秘密之前增加一个用户培训阶段。该训练阶段的目的是确保适当的说话音量，以分配有限的 MRR 预算，保证较高的掩蔽声音与语音比 (MSR) 和较高的语音与恢复噪音比 (SRR)。

在训练阶段，要求用户说出一个 3 位数的数字，以检查他们的说话音量。当训练录音的 MSR 大于 13 分贝且 SRR 大于 3 分贝时，用户将被视为训练成功。这一设置可确保人类语音的能量比掩蔽声音低 13 分贝。在我们的可用性研究中，用户平均只需要 1.6 轮就能通过训练标准。当要求用户当要求用户在 5 分钟后再次尝试相同的训练任务时，这一数字将降至 1.3。考虑到每次训练大约需要 5-6 秒，整个训练时间将少于 10 秒。大多数测试参与者表示，他们的语音很容易与我们当前的设置保持一致。本可用性研究的细节，如用户如何在训练后将语音保持在安全范围内，将在第 6 节中介绍。

4 安全分析

虽然 SafeChat 被设计为应用程序级的音频隐私泄漏防范工具，但要使 SafeChat 在现实环境中正常工作，仍存在一些要求。本节将介绍这些要求、目标威胁模型以及对 SafeChat 的潜在攻击。

4.1 威胁模型

安全聊天的目的是防止恶意软件未经授权的录音造成音频隐私泄露。尽管声音屏蔽可以防止通过其他渠道（如窃听）泄露隐私，但这超出了本文的讨论范围，因此也不是我们当前安全聊天设计的重点。

SafeChat 假定恶意软件和我们安装的应用程序具有相同的访问设备麦克风和扬声器的能力和权限。虽然 SafeChat 不需要修改设备操作系统，但它要求用户的设备既不能被root，也不能被破解。这一要求是必要的，因为被入侵的操作系统可以帮助恶意软件通过扬声器的音频链提取掩蔽声音，然后通过前面介绍的移除程序轻松移除。这是一个合理的假设，因为其他现有的音频隐私系统[13, 16, 17, 22, 29]也会在设备操作系统被入侵时被轻易攻破。

我们假定恶意软件在隐蔽录制音频后，可以通过语音识别引擎或类似恶意软件 *Soundcomber* [24]的众包方式轻松识别秘密。我们还假设恶意软件使用的语音识别引擎配备了噪音处理机制，并且恶意软件掌握了常见的音频预处理知识，如音源分离，从而方便其攻击被掩盖的秘密。安全聊天软件提供的安全性是指在上述假设条件下，恶意软件恢复屏蔽秘密的概率。

4.2 安全保证

其他基于密码学的安全系统可以提供精确的安全保证，如破解系统的预计时间，而 SafeChat 则不同，它不太可能提供这样的保证。不过，这也是非加密系统的常见问题，如 Wyner 的窃听通道 [28] 或其他物理安全无线系统。例如，大多数通过发送人工噪声 "掩盖 "数据包的无线安全系统 [11, 14]，其安全保证模型是确保窃听者接收到的数据包的信噪比小于数据包的编码/调制容量，因此窃听者从信息论上讲不可能恢复到用掩盖噪声接收到的数据包。

我们按照同样的分析思路来估算 SafeChat 所能提供的安全保证。具体来说，我们要评估授权方和被授权方之间私人/机密信息的信噪比差值。

未经授权的录音，即我们测量中的 MSR。然而，SafeChat 无法保证"编码/调制能力"，因为人类语音的调制方式与无线数据包不同。例如，语音信号中通常存在冗余信息，因此机器只需分析 MFCC 或多项式残余误差等统计特征就能理解人的语音。因此，我们仍然不知道 MSR 应该有多大。

在本文中，我们根据猜测秘密 x 的概率与知道其加密秘密 y 的概率无差（即 $P(x|y) = P(x)$）这一事实来评估 SafeChat 的隐私/秘密保护。

假设安全聊天知道恶意软件可以利用的最强大的语音引擎或

我们可以通过回答以下问题来定义安全聊天的安全性："恶意软件识别掩盖秘密的概率与纯随机猜测的概率之间的差异是多少？如果恶意软件无法使用假定的方法比随机猜测更好地猜测掩码秘密，那么 SafeChat 就达到了保密性。请注意，SafeChat 将噪音处理委托给了语音识别引擎和众包。我们还将讨论其他一些潜在的预处理方法，这些方法可能会帮助攻击者揭露秘密，但 SafeChat 对这些方法也有很强的抵御能力，下文和第 7 节将对此进行讨论。

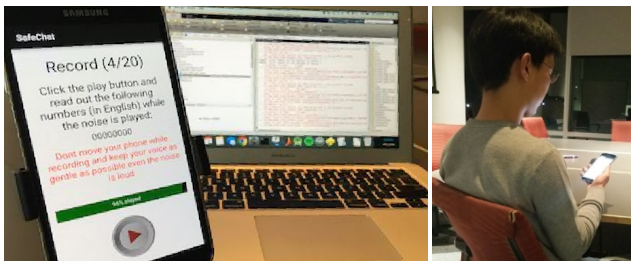
5 评估

我们将 SafeChat 作为安卓系统中的聊天应用程序和 Matlab 中的远程声音屏蔽服务器来实现。将 SafeChat 作为应用程序使用有助于自动将设备转为扬声器模式，并在需要时校准设备和语音音量。请注意，这种设计并没有牺牲太多 SafeChat 的优势。银行代表（Bob）仍然可以通过安装的聊天应用程序与客户（Alice）进行安全对话，而不是通过普通的远程通信。要求客户安装聊天应用程序也比要求手机制造商用新的系统补丁更新最新的音频隐私安全功能（如果有的话）要容易得多。

我们进行了实验来评估 SafeChat 的有效性。我们的评估旨在确定 SafeChat 是否能降低未经授权录音的语音智能，以至于恶意软件除了随机猜测外没有更好的方法来恢复被掩盖的秘密。正如第 4 节中讨论的威胁模型一样，我们关注的是恶意软件拥有利用人类思维或机器学习恢复秘密的知识/资源的情况。在不失一般性的前提下，我们在下面的评估中选择了一个 8 位数的掩码秘密，并以识别这个 8 位数秘密的准确率来衡量语音智能。之所以选择这个 8 位数字，主要是因为它是 TIDIGITS 数据集中最长的语料。我们的评估结果应适用于其他长度的信息，如 6 位数的 iPhone 密码或 9 位数的安全号码。

5.1 实验设置

我们在实验中收集了两个录音数据集，如图 6 所示，录音语音（秘密信息）是由测试设备附近的笔记本电脑扬声器播放的，或者是由参与者在类似位置说出的。笔记本电脑播放的语音是从 TIDIGITS 声音中的任意 8 位数录音中随机选择的。



(a) 笔记本电脑扬声器播放的语音 (b) 人类语音

图 6：实验设置。TIDIG-ITS 的语音由运行 SafeChat 的手机旁边的笔记本电脑播放。测试参与者被要求读出一个随机生成的 8 位数字。

数据集[18]，而测试参与者则随机读取一个 8 位数。

我们招募了 27 名参与者（17 名男性和 10 名女性）来建立人类语音数据集。其中 6 名参与者帮助录制了多次/多轮语音，而其他参与者仅录制了 6 次使用安全聊天软件最终设置的语音。例如，我们要求这 6 位参与者有意改变说话音量，即使音量超出了安全聊天的操作范围。笔记本电脑数据集简化了在不同场景下测试 SafeChat 的自动化过程，例如改变掩蔽音量、干扰增益或麦克风面板，而人类数据集则有助于我们了解 SafeChat 的实际性能。在实验和用户研究结束时，我们收集了 1600 多条录音，时间跨度超过 3.5 小时。这些录音后来被最先进的 Google Speech API [5] 或通过 Amazon Mechanical Turk [1] 招募的 317 名真实用户识别。

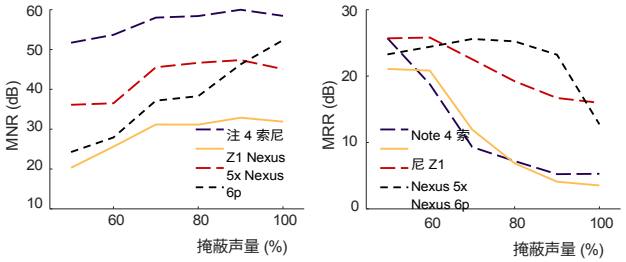
为了避免由于我们移除算法的语音假象而不是掩蔽声音导致识别准确率下降，谷歌语音 API 仅用于识别将笔记本电脑数据集中的掩蔽声音添加到 TIDIGIT 数据集中的随机语音中的录音。这种设置有利于提高攻击性能，因为它假定攻击者掌握了信号预处理知识，可以完全消除语音伪迹。我们还尝试使用我们的数据集（即屏蔽语音）重新训练一个开源语音识别引擎，但由于攻击性能比使用最先进的 Google Speech API 差很多，因此省略了这一结果。对使用其他语音识别引擎的攻击性能进行鉴定是我们未来工作的一部分。

5.2 屏蔽声音及其去除的效果

我们首先从 MNR 和 MRR 的角度评估了播放和移除掩蔽声音的效果。为了展示不同设备的能力，我们在表 1 所示的 6 种不同设备上应用了我们的校准过程，同时将设备扬声器音量从 50% 变为 100%。图 7 显示了在 6 种设备中的 4 种设备上添加/去除掩蔽声音的结果。

设备	ch_{calib}	vol_{calib}(%)	MNR(dB)	MRR(dB)
银河 S4	2	70	35	18
Galaxy S5	1	70	32	20
Galaxy Note4	1	50	51	26
索尼 Z1	1	60	24	21
Nexus 5X	1	60	38	26
Nexus 6P	1	80	39	26

表 1：安全聊天在不同设备上的校准设置



(a) MNR (b) MRR

图 7：掩蔽声音在不同设备上的效果。当扬声器音量较大时，一些设备的屏蔽性能较好，而如果音量过大，其他设备则无法有效消除屏蔽声。(为了便于阅读，Galaxy S4/5 的结果略去不提）。

在校准过程中，我们发现所有测试设备的最大扬声器音量都能产生高于 30 分贝的 MNR，这表明在适当的设置下，扬声器的音量足以掩盖语音。在消除掩蔽声音的效果方面，大多数设备在 50-80% 的扬声器范围内表现出较好的消除效果，在我们测试的设备中，50% 的扬声器音量通常可以达到 20dB 以上的 MRR。当音量根据我们的校准算法进行设置时，安全聊天软件能够从录制的音频中去除高达 26 分贝的掩蔽声音。

请注意，校准行为因设备而异。例如，如图 7(a)所示，当 Note 4、Nexus 5X 和索尼 Z1 的 MNR 在扬声器音量增加超过 80% 时停止增加时，Nexus 6P 在全音量播放掩蔽声音时仍能有高出 10dB 的 MNR。这种现象可能是由于设备中配备的麦克风和扬声器的动态范围不同造成的。不过，尽管 Nexus 6P 的动态范围似乎足够高，可以全音量播放掩蔽声音，但如前所述，全音量播放声音会导致非线性失真和麦克风饱和，从而使 SafeChat 无法有效消除掩蔽声音。例如，如图 7(b)所示，当掩蔽声音的音量从 80% 增加到 100% 时，Nexus 6P 的 MRR 从 26dB 下降到 13dB。其他设备也有相同的表现，因此我们的校准算法将它们的扬声器音量设置在 50% 到 80% 之间，以便在 MNR 和 MSR 之间取得最佳平衡。在使用附近笔记本电脑播放语音的实验中，Nexus 6P 而 Galaxy S4 只能在播放笔记本扬声器音量的 70% 时隐藏语音。表 1 总结了校准设置的详细信息。

5.3 屏蔽声音去除的鲁棒性

用户移动和环境噪声可能会影响我们的掩蔽声音去除过程。例如，我们发现 Nexus 6P 的 MRR 如表 1 所示

当用户将坐姿改为走姿时，声音从 26 分贝下降到 23 分贝，但 MRR 仍然很高，足以支持 port SafeChat。原因是估计的声音响应响应（$H_{(s)}(*)$）是由播放的遮蔽声音如何决定的。

在麦克风中合并/接收声音，设备的移动会改变声音传输的行为。例如，大幅度摇晃手机会导致 10-15 分贝的性能下降，但这并不是手机用户的典型行为。

SafeChat 对普通环境噪声有很强的适应能力。例如，在类似图 6（a）的环境中，当笔记本电脑播放响亮的摇滚音乐时，仅观察到 3dB 的 MRR 下降。我们还在拥挤的学生活动中心的咖啡厅附近收集了 21 位参与者的数据，但没有观察到明显的性能下降。常见的环境噪音不会影响掩蔽声音的效果。

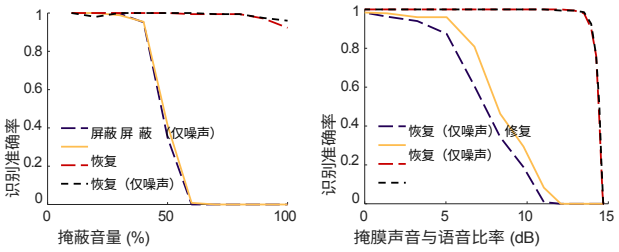
因为环境噪声与掩蔽声音并不相关，所以声音去除率很低。大音量的背景噪声实际上有助于隐藏口语秘密。

在我们的实验中，环境噪声导致的一个具体问题是 MSR 被低估。例如，由于测试地点附近其他人的大声笑声，语音能量包络可能会激增，从而通过使用这种错误的语音信号作为参考，降低了估计的 MSR。在这种情况下，即使掩蔽声音大到足以隐藏秘密，用户也无法通过训练。解决这一问题的方法之一是要求目标接收器检查训练失败的恢复信号，并确定是由于所说的秘密还是环境噪声造成的。

尽管安全聊天可能会出现其他一些极端情况，但应该注意的是，安全聊天仅在秘密对话期间触发。用户不太可能在突然移动电话或在非常嘈杂的环境中进行私人/秘密对话。如果需要，预期接收方也可能会要求用户找一个安静的环境说秘密。我们的实验表明，SafeChat 能够适应典型的现实生活环境，例如在典型的公共场所边走边说。

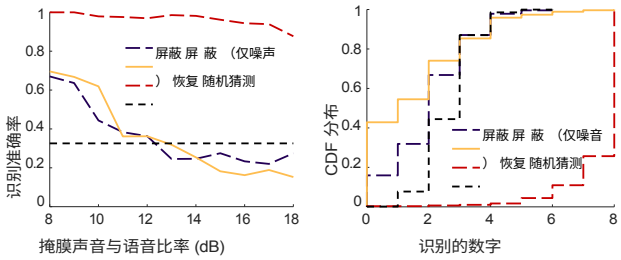
5.4 针对谷歌语音 API 的攻击性能

如前所述，我们考虑的威胁模型是恶意软件可以尝试使用最先进的语音识别引擎从屏蔽录音中识别出屏蔽的秘密。具体来说，我们使用谷歌语音应用程序接口（Google Speech API）来识别不同遮蔽声音设置下的笔记本电脑痕迹。请注意，输入谷歌语音应用程序接口的录音没有经过预处理，因为该应用程序接口已经设计用于处理嘈杂的音频，而降噪预处理会降低识别准确率。这种内置的噪音处理方法有助于恶意软件揭开被掩盖的秘密。



(a) 掩膜音量变化 (b) 不同语音音量

图 8：谷歌语音 API 的识别准确率。掩码音频和恢复音频之间的准确率差距很大，这说明音频的有效性很高。



(a) 不同语音音量 (b) 匹配 CDF 的数字

图 9：与人类对比的识别准确率。75%的重新掩盖的秘密可以被正确识别，而参与者（人类）无法完全识别任何被掩盖的录音。

我们使用标准的单词识别准确率来评估 SafeChat 的性能。单词识别准确率的定义为 $(N-D-S-I)/N$，其中 N 为掩码中的数字位数。语音，I 和 D 是识别文本中添加或遗漏的错误数字。表示识别文本中添加或遗漏的错误数字。S 表示有多少数字被错误数字替换。Galaxy S5 在不同实验设置下的识别准确率结果如图 8 所示。

图 8(a)显示了在掩蔽声音中加入低 13 分贝的语音，然后在 Galaxy S5 上录制的结果，即 MSR 固定为 13 分贝。随着播放的掩蔽声音音量增大，识别准确率也随之降低。当扬声器音量设置大于 50%时，屏蔽语音的识别准确率急剧下降，而恢复的语音仍能被谷歌语音 API 高概率识别。这种巨大的性能差距体现了 SafeChat 在授权录音和未授权录音之间提供差异信息的有效性。图中还显示了仅有遮蔽噪声的遮蔽声音与同时有遮蔽干扰和噪声的遮蔽声音之间的差异，但这种差异并不十分明显，因为机器学习算法通常是通过统计特性（如 MFCC 或多项式残差误差）来识别语音的，而添加结构化干扰（如音量较低的另一个人的声音）并不会对这些特性产生太大影响。当声音由人类识别时，添加干扰的效果会更加明显。

在了解了不同掩蔽音量下的性能变化后，图 8(b) 显示了以说话者音量的 70%（即 Galaxy S5 中的校准设置）播放掩蔽音并添加不同语音音量时的识别准确率。本实验的目的是了解使用 SafeChat 可以保护的语音音量水平。如图所示，Galaxy S5 防止秘密被 Google Speech API 恢复的最佳操作范围是 MSR 大于 11 分贝且小于 15 分贝。在这一设置下，谷歌语音应用程序接口识别未屏蔽语音的准确率可达 98%，而理解屏蔽语音的准确率则低于 0.1%。需要注意的是，这一操作范围因设备而异，但 SafeChat 一般会将 MSR 的下限设为 12-15dB，以防止机密信息泄露。此功能的可用性我们将在第 6 节中进一步讨论这些设置，例如人们如何轻松地保持适当音量的声音。

5.5 针对人类识别的攻击性能

为了评估 SafeChat 在人类识别方面的性能，我们通过 Amazon Mechanical Turk 招募了超过 317 名用户来识别 6 名测试参与者录制的音频。由于这项用户研究不会干扰人类行为，也不会侵犯用户隐私，因此被我校认定为不受 IRB 监管。我们故意告诉用户：(i) 要识别的内容是一个 8 位数的数字；(ii) 要识别的数字可能隐藏在噪音中。我们允许用户随意重复识别每段录音，并多次编辑他们的答案。这种设置的目的是模仿恶意软件试图通过众包方式恢复秘密的情景。

图 9(a) 显示了在 Nexus 6P 上录制的音频在校准设置下播放遮蔽声音时的识别准确率。如图所示，一旦 MSR 大于 13dB，用户感知到的语音智能就会被认为不过是随机猜测一个 8 位数字的结果。参与者识别掩蔽声音的准确率一般高于谷歌语音应用程序接口，因为后者在识别无法识别的掩蔽声音时往往会返回一个空字符串，而参与者已经知道有一个隐藏的数字，通常会返回一些（甚至是错误的）答案。

另一方面，恢复录音中的秘密识别准确率可达 95%。当 MSR>17dB 时，性能开始下降，因为残留的掩码噪声可能会阻碍用户识别一些低音量语音信号（如 SRR<3dB）。这种掩码识别和秘密恢复之间的巨大准确率差距与之前使用谷歌语音 API 的结果一致，从而证明了 SafeChat 能够有效地隐藏秘密，避免被语音识别引擎和人类恢复。根据上述结果，我们将用户语音的 MSR 和 SRR 门限分别设置为 13dB 和 4dB。

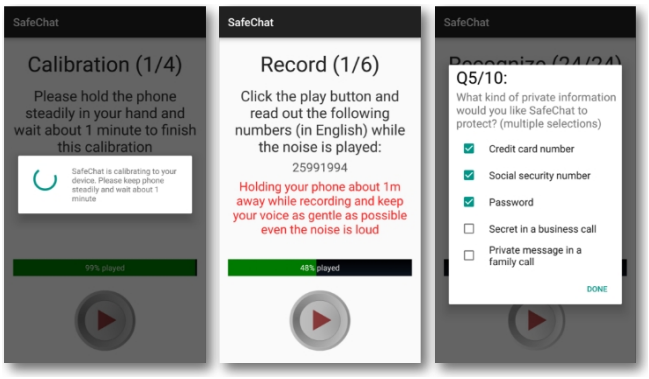


图 10：用户研究应用程序界面。用户完成自我校准阶段后，需要通过培训阶段，录制多个音频片段，然后填写调查问题。

图 9(b)显示了在最终设置下数字被识别的 CDF。超过 75% 的恢复录音都能被目标接收器完全识别，而恶意软件则无法识别任何被屏蔽的音频。此外，超过 95% 的恢复录音的数字错误少于 2 个。由于用户在识别电话语音时通常会出现 5%或更高的单词错误率[21]，因此通过安全聊天工具进行通话不会给目标接收者带来比普通语音更多的负担。大多数被识别出有 1-2 位数错误的机密都被其他用户正确识别。在评估安全聊天的性能时，多数票决定 [19] 可以进一步避免这些用户错误。考虑到目标接收者通常会在无法识别秘密时要求用户重复该秘密（即使没有安全聊天软件也是如此），这种准确性足以支持大多数用例。假设每次录音都是独立于其他录音的，那么我们的结果可以解释为，超过 98% 的 8 位数秘密可以在重复 2 次内完全恢复。

在比较屏蔽干扰（即从现有数据库中添加口语句子）的效果时，发现用户更有可能正确识别 8 位数中的 7 位数。

只包含掩蔽噪声的录音，即 m_n。这种情况通常发生在用户的语音音量没有被掩蔽噪声完全掩盖的角落。在这种情况下，添加的干扰（即 m_i）会使恶意软件对多个语音源同时存在感到困惑，从而帮助 SafeChat 防止这种泄漏。如图 9(b)所示，通过同时添加屏蔽干扰和噪声，安全聊天可以确保恶意软件猜测屏蔽秘密的概率不高于随机猜测。请注意，在有干扰的屏蔽录音中，用户正确识别 2 个（共 8 个）数字的概率较高，这是因为人类会错误识别添加了干扰的数字，而在只有屏蔽噪音的录音中则没有这种干扰。

6 可用性研究

本可用性研究的目的是评估 (1) 用户是否能在安全聊天的操作范围内轻松控制自己的语音音量，(2) 用户在阅读私人/机密信息时是否能忍受所发出的噪音，以及 (3) 考虑到安全聊天保护音频隐私的目的，用户是否愿意在受到一些使用限制的情况下使用它。

为了回答这些问题，我们请 27 位参与者试用了演示应用程序，如图 10 所示。请注意，本可用性研究是在首次使用 SafeChat 的情况下进行的（少数参与者随后被要求参加了其他测试，测试中的录音量各不相同）。我们首先介绍了使用噪音来保护智能手机上的私人/秘密信息的想法，并让他们试用了安全聊天的演示应用程序。演示应用程序的配置是，首先要求用户通过第 3 节所述的培训阶段，然后要求用户在启用安全聊天功能的情况下完成 3 个读取随机 8 位数字的任务。每当录音超出安全聊天的操作范围，即（MSR < 13dB 或 SSR < 4dB），系统就会弹出提示消息，要求用户用更大或更小的声音重复相同的任务。用户完成测试（即 1 次训练和 3 次录音）后，会被要求填写一份调查问卷。填写完调查问卷后，参与者会被要求再次试用相同的演示应用程序。第二次测试是为了确定用户是否能更轻松地满足安全聊天的要求，因为他们现在已经知道如何使用安全聊天。

与仅仅要求参与者保持安静相比，指示参与者以某种姿势录音会有更高的几率录制出与安全聊天兼容的语音。根据我们的初步实验，人们说话的声音通常比 SafeChat 预期的要大，而且当他们被告知要小声说话时，他们倾向于把麦克风靠近嘴边。当只发出 "小声说话 "的指令时，人的这种本能会使记录的秘密变得更大声（因为靠近麦克风说话）。与此相反，要求用户以某种姿势握住手机则更容易遵循，尤其是在向他们展示了所需姿势的图片时。具体来说，我们要求所有参与可用性研究的人以类似图 6(b) 所示的姿势录制语音。保持这种姿势不仅可以减少录音的秘密音量，还可以减少对听到掩蔽声音的干扰。

根据上述结果，用户在首次使用安全聊天时需要 1.6 轮才能通过初始训练阶段。考虑到每次训练大约需要 6 秒（包括处理时间），训练的总体时间开销平均不到 10 秒。通过训练阶段后，用户只需 1.3 个回合就能读取一个 8 位数的密文，且音量符合安全聊天的设置。重复次数的减少表明，训练能有效地引导用户学习正确的语音音量。请注意，我们目前只在录制完整语音后才验证语音标准。考虑到我们计划进行的改进，即实时监控语音音量并在数字说得太大声时禁用对话，满足重新要求的开销可以进一步降低。70% 的测试参与者能在一轮内完成语音，30% 的参与者需要多一轮才能完成。此外，随着用户对安全聊天的熟悉，正确发言所需的回合数也会进一步减少。例如，当用户进行第二次 8 位数录音时，85% 的用户第一次尝试就能完成，而在填写调查问卷后，用户平均只需要 1.3 轮就能再次通过训练。

在我们的可用性研究中，录音的性能与上一节所示的类似。识别屏蔽录音和恢复录音的准确率分别为 22% 和 93%。根据我们的调查，18 位用户认为语音音量要求很容易达到，24 位用户可以忍受发出的噪音（本研究校准音量为 80%）。为了防止未经授权的录音泄露隐私，有 22 位 用户愿意使用 SafeChat，尤其是在电话中告知信用卡信息或商业秘密时。当得知应用程序可以在后台录音时，大多数参与者都感到惊讶。极少数参与者希望语音音量阈值能更高一些，而大多数人都能根据当前设置调整音量。只有两名与会者抱怨声音被掩盖了。一位与会者留言说，这项技术非常有用，实际上也可以应用到语音信息录制应用程序中。考虑到我们目前的设计需要用户通过安全聊天工具拨打电话，当用户录制秘密语音信息时，安全聊天工具似乎更适合用来防止音频隐私泄露（因为他们无论如何都需要一个应用程序来这样做）。

7 讨论

我们提出了 SafeChat，以减少新出现的通过恶意软件未经授权录音而泄露音频隐私的情况。SafeChat 无需对手机操作系统进行任何修改即可保护智能手机中的音频隐私。我们的评估表明，基于假定的威胁模型，SafeChat 是有效的。具体地说，面对最先进的谷歌语音应用程序接口（Google Speech API）和 371 名在线招募的用户，安全聊天软件能够有效地隐藏和恢复秘密对话。关于 SafeChat 的一个常见问题是："在识别被掩盖的秘密之前应用低通滤波器如何？我们的测试结果表明，由于添加的掩蔽噪声覆盖了整个人类语音频谱，因此安全聊天可以抵御这种低通攻击。基于 "盲 "声源分离的类似潜在攻击，如独立分量分析（ICA），如果录音麦克风的数量大于播放扬声器的数量，理论上可能能够消除掩蔽声音。然而，我们经常注意到，在现实中，除了用于恢复安全信息的麦克风外，SafeChat 会使所有麦克风达到饱和，因此 ICA 无法分离 "独立 "声源。分析和评估对其他潜在攻击（如果有的话）的安全保证是我们未来工作的一部分。

安全聊天的另一个潜在问题是，手机的回声消除功能可能会将掩蔽信号视为回声，然后自动取消它们。需要注意的是，我们在实验中已经打开了安卓系统的噪声抑制器和回声消除器，但并没有注意到这种消除的发生。不过，在我们在线招募了另外 78 名测试参与者在他们的设备上安装 SafeChat 之后，我们发现了一种特殊情况，即在 LYF Flame 1 中的应用程序接收到遮蔽声音之前，设备操作系统或硬件已经将其消除。由于这种情况可以通过我们的校准算法轻松识别，因此安全聊天可以通知/警告用户，它无法支持用户的删除操作。在进一步检查了 Nexus 6P 的源代码后，我们发现当用户启用以下功能时也可能出现类似现象

首先使用 SafeChat，然后拨打/接听电话（通过电信）。在这种情况下，系统服务（即 CallManager）拥有比 SafeChat 更高的权限来更改音频链并启用回声消除。请注意，回声消除是在芯片级实现的，即高通公司专有的 *Fluence* 语音增强技术[9]。在我们的实验中，普通应用 API（如 AcousticEchoCanceler）无法成功触发该功能。相反，可以通过修改 pers.audio.fluence.voicecall 来控制该功能。遗憾的是，SafeChat 现在无法自动停用该功能，因为它会重新要求系统许可。未来的一个可能方向是要求手机制造商提供适当的 API 来控制低级回声消除功能。这种机制对于安全聊天以外的其他用途可能是必要的，因为在许多使用案例中，回声消除功能可能会被错误触发。请参阅论坛上关于在 Android 中禁用回声消除功能的讨论 [2]。我们正在研究的另一个短期解决方案是在使用安全聊天时阻止远程通信，或将安全聊天作为一项系统服务来实现。

8 结论

在本文中，我们展示了使用声音屏蔽来阻止恶意软件未经授权的录音造成隐私泄露的可行性。我们设计、实现并评估了 SafeChat，它是安卓聊天应用程序和远程声音屏蔽服务器的新型组合，可以在不修改现有操作系统中的音频隐私/许可方案的情况下为移动设备提供音频隐私。我们的广泛实验评估表明，SafeChat 能够在授权和非授权录音之间实现高达 26 分贝的信号强度差异。通过适当的设置，这种信号强度差可以阻止人们或最先进的语音识别算法理解被遮蔽的声音，而授权应用程序则可以通过我们的遮蔽声音消除算法理解大部分隐藏的语音。我们的可用性研究参与者支持上述发现，他们中的大多数人都希望使用 SafeChat 来保护他们的私人信息，如信用卡号或密码。

参考文献

[1] [n.d.].Amazon Mechanical Turk. https://www.mturk.com/mturk/welcome.

[2] [n.d.]. 安卓 麦克风 噪音 消除） 问题。https://forums.oneplus.net/threads/fixed-for-good-microphone-issue.222059/。

[3] [n.d.]. 剑桥 声音 屏蔽 系统。http://www.soundmasking.com/selectmasking.html.

[4] [n.d.].FlexiSpy 应用程序。https://www.flexispy.com/en/mobile-and-cell-phone-spy-features.htm。

[5] [n.d.].谷歌语音 API。https://cloud.google.com/speech.

[6] [n.d.].MobiStealth 应用程序。http://www.mobistealth.com/features.php.

[7] [n.d.].Soft dB 声音屏蔽系统。http://www.softdb.com/sound-masking/。

[8] [n.d.].TheTruthSpy app. http://thetruthspy.com/features/. [n.d.].

[9] [n.d.]. 宽 带 声 码 器 和 Fluence TM 降 噪 - 高 通 公 司 。https://www.qualcomm.com/media/documents/files/hd-voice.ppt。

[10] 2008.使用衔接指数客观测量 开放式办公室内语音隐私的标准测试方法。*ASTM International* (2008)。

[11] N.Anand, Sung-Ju Lee, and E. W. Knightly.[n.d.].STROBE：使用零强迫波束成形主动确保无线通信安全。*IEEE INFOCOM '12 论文集*。720-728.

[12] A.Benyassine, E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit.1997.ITU-T Recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications.*IEEE Communications Magazine* 35, 9 (1997), 64-73.

[13] Soteris Demetriou、Xiao-yong Zhou、Muhammad Naveed、Yeonjoon Lee、Kan Yuan、XiaoFeng Wang 和 Carl A Gunter。[n.d.]。您的加密狗和

银行账户里有什么？安卓外部 资源的强制和自由保护。In *NDSS '15*.

[14] S.Goel and R. Negi.[n.d.].使用人工噪声保证保密。*IEEE Trans.Wireless.Comm.* ([n. d.]), 2180-2189.

[15] Monson H Hayes.2009.*统计数字信号处理与建模*》。John Wiley & Sons.

[16] Stephan Heuser, Adwait Nadkarni, William Enck, and Ahmad-Reza Sadeghi.[n.d.].ASM: A Programmable Interface for Extending Android Security. *USENIX SEC'14 论文集*。1005-1019.

[17] Peter Hornyack、Seungyeop Han、Jaeyeon Jung、Stuart Schechter 和 David Wetherall。[n.d.].这些不是你要找的机器人：改造安卓系统以保护数据免受不正当应用的侵害。*ACM CCS '11 论文集*。639-652.

[18] R. 加里 - 伦纳德和乔治 - 多丁顿。[n.d.].TIDIGITS Dataset. https://catalog.ldc.upenn.edu/ldc93s10.

[19] Matthew Marge, Satanjeev Banerjee, and Alexander I. Rudnicky.[n.d.]. 使用 Amazon Mechanical Turk 转录和注释会议发言，用于 Extractive Summarization。In *Proceedings of CSLDAMT '10*.99-107.

[20] Rajalakshmi Nandakumar、Krishna Kant Chintalapudi、Venkat Padmanabhan 和 Ramarathnam Venkatesan。[n.d.].Dhwani：安全点对点声学 NFC。*ACM SIGCOMM '13 论文集*。63-74.

[21] Scott Novotney 和 Chris Callison-Burch。[n.d.]。便宜、快速、足够好：自动语音识别与非专家转录。*HLT '10 论文集*。207-215.

[22] Giuseppe Petracca、孙玉琼、Trent Jaeger 和 Ahmad Atamli。[n.d.].Au-Droid：防止对移动设备音频通道的攻击。In *Proceedings of ACM ACSAC '15*.181-190.

[23] Nazir Saleheen、Supriyo Chakraborty、Nasir Ali、Md Mahbubur Rahman、Syed Monowar Hossain、Rummana Bari、Eugene Buder、Mani Srivastava 和 Santosh Kumar。[mSieve：移动传感器数据时间序列中的差异行为隐私。In *Proceedings of ACM UbiComp '16*.706-717.

[24] Roman Schlegel、Kehuan Zhang、Xiao-yong Zhou、Mehool Intwala、Apu Kapadia 和 XiaoFeng Wang。[n.d.].Soundcomber：用于智能手机的隐秘和感知上下文的声音 木马。在 *NDSS '11*.

[25] Souvik Sen、Naveen Santhapuri、Romit Roy Choudhury 和 Srihari Nelakuditi。[n.d.].连续干扰消除：后发视角。In *Proceedings of the 9th ACM Hotnets '10*.17:1-17:6.

[26] Manish Shukla、Purushotam Radadia、Shirish Subhash Karande 和 Sachin Lodha。[n.d.].DEMO：使用热补丁实时屏蔽信用卡声音。*ACM CCS '13 论文集*。1351-1354.

[27] Yu-Chih Tung、Kang G. Shin 和 Kyu-Han Kim。[n.d.].针对基于链路的数据包来源识别的模拟中间人攻击。In *Proceedings of ACM MobiHoc '16*.331-340.

[28] A.D. Wyner.1975.The wire-tap channel.*The Bell System Technical Journal* 54, 8 (1975), 1355-1387.

[29] Zhi Xu 和 Sencun Zhu.[n.d.]。SemaDroid：智能手机的隐私感知传感器管理 框架。In *Proceedings of ACM CODASPY '15*.61-72.

[30] B.Zhang、Q. Zhan、S. Chen、M. Li、K. Ren、C. Wang 和 D. Ma. 2014.PriWhisper：为智能手机实现无密钥安全声学通信。*IEEE Internet of Things Journal* 1, 1 (2014), 33-45. https://doi.org/10.1109/JIOT.2014.2297998

[31] N.Zhang, K. Yuan, M. Naveed, X. Zhou, and X. Wang.[n.d.].Leave Me Alone：安卓系统上针对运行时信息收集的应用级保护。In *2015 IEEE Symposium on Security and Privacy*.915-930.