VeriSMS: A Message Verification System for Inclusive Patient Outreach against Phishing Attacks

Chenkai Wang
chenkai3@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, IL, USA

Zhuofan Jia
zhuofan7@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, IL, USA

Hadjer Benkraouda
hadjerb2@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, IL, USA

Cody Zevnik
cody.zevnik@osfhealthcare.org
OSF Healthcare
Peoria, IL, USA

Nicholas Heuermann
nicholas.heuermann@osfhealthcare.org
OSF Healthcare
Peoria, IL, USA

Roopa Foulger
roopa.foulger@osfhealthcare.org
OSF Healthcare
Peoria, IL, USA

Jonathan A. Handler
jonathan.a.handler@osfhealthcare.org
OSF Healthcare
Peoria, IL, USA

Gang Wang
gangw@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, IL, USA

ABSTRACT

Patient outreach enables timely communication between patients and healthcare providers but is vulnerable to phishing/spoofing attacks. In this paper, we work with a U.S.-based healthcare provider to design an inclusive method to address this threat. We present *VeriSMS* which allows patients to call a voice agent to verify whether the received (sensitive) messages are indeed sent by their healthcare provider. We design the system to be inclusive: it is accessible to patients who only have access to SMS and phone call capabilities. We perform a two-part user study to refine the system design (N=15) and confirm users can correctly understand the system and use it to identify spoofed/phishing messages (N=35). A key insight from our study is to not exclusively optimize for strong security but to tailor the designs based on user habits. Our result confirms the effectiveness and usability of VeriSMS and its ability to significantly increase adversaries' costs.

CCS CONCEPTS

• **Security and privacy → Usability in security and privacy**.

ACM Reference Format:
Chenkai Wang, Zhuofan Jia, Hadjer Benkraouda, Cody Zevnik, Nicholas Heuermann, Roopa Foulger, Jonathan A. Handler, and Gang Wang. 2024. VeriSMS: A Message Verification System for Inclusive Patient Outreach against Phishing Attacks. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3613904.3642027

1 INTRODUCTION

Patient outreach enables timely communication between patients and healthcare providers. Well-designed patient outreach programs can extend the reach of healthcare through health education, basic health screening, and enabling access to key services, which improves the overall health outcomes of individuals and the community [8, 67, 74]. However, patient outreach channels can be vulnerable to security attacks, in particular, *phishing attacks*, which often cause financial loss, sensitive information leakage, and more importantly, mistrust of the outreach channel that ultimately undermines the value of the patient outreach programs. According to the Federal Trade Commission (FTC), healthcare is among the top 10 fraud categories, reporting 68,496 cases with a total of $17 million loss in 2022 [18].

During phishing attacks, attackers often impersonate the identity of health providers via *spoofing* techniques [26, 78] to gain the victim's trust. This further lures the victim to give away sensitive information or perform damaging actions [69, 83]. Such spoofing technique is available to attack various communication channels such as email (via email address spoofing [26]) as well phone calls and short message services (SMS) (via caller ID spoofing [64, 70, 78]). The vulnerability is rooted in the lack of sender authentication in existing communication systems and protocols. Unfortunately, it is difficult to quickly fix them due to the common concerns for major disruptions and increased costs for upgrading the underlying systems and hardware. The efforts of designing, promoting, and integrating secure protocols into existing systems have been made for several decades [26, 70].

In this paper, we aim to provide an *inclusive* solution to address phishing and spoofing concerns in patient outreach programs, by collaborating with a U.S.-based healthcare provider OSF Healthcare. We focus on the SMS-based communication channel because not all patients, especially older adults, own smartphones that support sophisticated mobile apps and security features (at least for OSF

CHI '24, May 11–16, 2024, Honolulu, HI, USA

Healthcare's patient population). This is in line with recent surveys reporting that only 61% of older adults in the U.S. own a smartphone [16] and non-smartphones are still commonly used by older adults [2, 57]. As such, the outreach program at OSF Healthcare is primarily focused on SMS.

To combat spoofing/phishing attacks, we present *VeriSMS*, a message verification scheme that allows patients to check if the received (sensitive) messages are indeed sent by their healthcare provider. To minimize the cost, we piggyback on existing mechanisms in healthcare systems to design the "root-of-trust" as the physical card that patients obtain during in-person visits to healthcare facilities, which carries the healthcare provider's true phone number. The outreach messages are designed to contain a random *Message ID* and a pair of English words as the *Secret Words* to support their verification. Upon receiving a message (e.g., that demands critical actions from the patient), the patient can call the phone number on the physical card to interact with an *voice agent* hosted by the healthcare provider, which will guide the patient to verify their messages. We carefully tailor the design of the Message ID (as dynamic/random numbers) and Secret Words (as static English words) to balance security and usability, while ensuring *bidirectional authentication* between patients and healthcare providers during the verification process. Our security analysis (Section 4.3) confirms that VeriSMS can significantly increase the cost of attackers (by 7–12 orders of magnitude) while staying resilient to adaptive attacks.

We perform a two-part user study to examine whether users can correctly understand the system and use it to identify spoofing messages. It contains (1) an exploratory study (N=15) to identify issues in the initial system design, and (2) a validation study (N=35) to confirm the effectiveness of the revised system. While *VeriSMS* is designed for all users, we particularly want to ensure users who are more dependent on the phone call and SMS functionality (e.g., older adults) can use the system properly. As such, for the validation study, we intentionally oversampled older adults (43% of the participants are over 55 years old). To improve the realism of the study, participants use their *personal mobile phones* to interact with VeriSMS.

Throughout the two studies, we find that participants can generally correctly use the system to verify the messages: all participants are able to identify the spoofing messages under different conditions. Meanwhile, we did observe a few users making occasional/rare mistakes on benign messages (e.g., false positives) when users made their decisions solely based on the message content (rather than verification with the voice agent). During the user study, we intentionally introduced a 7-day gap, inviting a subset of participants for another test, which confirms that users can still remember how to use the system after the time gap. The System Usability Scale (SUS) score of VeriSMS has a mean of 79.1 (with a median of 85), indicating "good" to "excellent" usability.

A key insight from our study is that VeriSMS *does not exclusively optimize for strong security*, but aims for practicability by tailoring the designs based on *user habits.* For example, VeriSMS uses two *static* English words as a user's "Secret Words" rather than using dynamically changing random numbers (which would have been more secure). The reason is that this more secure version would not work in practice because, as observed in the user study, most participants do not call to *verify every single message.* The static Secret Words design of VeriSMS guarantees a basic level of security even if certain users do not call everytime.

Given the positive result from the user study, our partners at healthcare provider OSF Healthcare has expressed interests in performing internal tests on certain patients groups.

This paper makes three key contributions.

- We propose VeriSMS to defend against spoofing and phishing attacks during patient outreach. The system is designed to be *inclusive*: it does not require extra hardware/setup and only need SMS and phone call capability that is widely available on both smartphones and feature phones.
- We perform a security analysis on the proposed method to illustrate the increased costs of attackers and the basic system resilience against adaptive attacks.
- We revise and validate the system design based on user studies. We confirm the effectiveness and usability of the system and discuss future improvements for practical deployment.

2 BACKGROUND AND RELATED WORK

2.1 Patient Outreach

Patient outreach is critical to ensure patients get timely medical attention by staying in touch with caregivers [27, 42]. For example, a recent study demonstrated success in improving mammography screening for women and reducing inequity in cancer screening in a disadvantaged population using a set of outreach solutions that included SMS messaging [74]. However, the design of an outreach channel needs to proactively consider security risks. A key threat faced by such channels is phishing attacks [26, 37, 64, 94]. The concern is not only about the financial loss and emotional distress that phishing can cause but also about the risk of compromising the effectiveness of the outreach channel, preventing the health system from achieving the target clinic effect (e.g., immediate cancer screening for the patients).

2.2 SMS-Based Phishing and Spoofing

Spoofing is a common technique used by phishing attackers to make their communication appear legitimate by impersonating trusted parties [26, 77]. SMS spoofing which allows attackers to display the phone number (i.e., caller ID) of a trusted part on their phishing messages [28], which is prevalently used in practice [18]. Caller ID spoofing vulnerability [22, 77] is rooted in the lack of caller ID authentication in the mobile network, more precisely, the Signaling System 7 (SS7) protocol [61, 78]. Various online services can provide caller ID spoofing functionality, via the service provider's IP-PSTN[1] gateway connections. Adversaries can perform spoofing by joining the mobile network as a small carrier or a third-party service provider (e.g., for Voice-over-IP, or VoIP) [70]. Alternatively, SMS spoofing can be done by establishing a false base station or cell site simulator (CSS) to send SMS to nearby victim users [51, 73]. The former method is more practical given it can be done *remotely* while the latter requires getting physically close to the victims.

[1] PSTN stands for "Public Switched Telephone Network."

2.3 Existing Solutions and Limitations

Most existing works on anti-phishing or anti-spoofing are focused on phishing emails [7, 20, 24–26, 32, 45, 63, 81], phishing websites [11, 23, 36, 40, 52, 53, 59, 60, 82, 92, 93] or phishing URLs [15, 39, 68, 75]. To prevent or detect SMS-based spoofing, particularly, caller ID spoofing, the most fundamental solution is to modify/redesign the existing mobile network to add caller ID authentication [14, 50, 62]. For instance, protocols such as Secure Telephone Identity Revisited (STIR) and Signature-based Handling of Asserted Information Using toKENs (SHAKEN) [17, 62] have been proposed to address the caller ID spoofing problem. A practical challenge is that carriers may be reluctant to deploy the STIR/SHAKEN due to the implementation overhead and the cost of purchasing/deploying new hardware. Recently, FCC [17] started to enforce STIR/SHAKEN for small carriers, which is a promising development. However, there are many mobile carriers out of the scope of FCC's jurisdiction (e.g., especially malicious actors). In this paper, we still assume phone numbers can be spoofed (which is still the reality) and focus on solutions that do not involve modifying existing mobile networks/infrastructure.

A line of related work is focused on detecting and blocking spam phone calls (or robocalls) [38, 56, 65, 77] or SMS [46, 90]. These techniques require either building a machine learning (ML) classifier based on call log datasets [38] and SMS traces [90] or training a personal assistant (an ML model) to interrupt/interact with malicious callers [56]. However, these solutions typically require accessing large-scale *sensitive* datasets of call logs/SMS traces. More critically, they require users to own a smartphone and install their corresponding smartphone apps, which may not be feasible for socioeconomically disadvantaged patients.

3 DESIGN GOALS AND THREAT MODEL

To secure the SMS-based patient outreach channel, we first list the key design goals and challenges.

Requiring limited Initial setup. Not all patients have explicitly signed up with the healthcare provider's outreach program to establish a secure channel before the communication. As a result, healthcare systems often need to send unsolicited messages, and patients need to verify the sender with no (or limited) initial setup.

Low-cost. As an inclusive system, it should be accessible to patients of low socioeconomic status. Solutions that require purchasing extra hardware are not considered. For example, hardware-based solutions such as Time-Based One-Time Password (TOTP) [49] or HMAC-Based One-Time Password (HOTP) [48] can be repurposed to verify the received message. However, such hardware devices typically cost $8–$30 (as of 2023) [19, 66]. Additionally, they need a secure enrollment step (which violates the first design goal above).

Easy to Use. Most patients are not trained to recognize phishing attacks. This is especially true for older adults [54, 71]. The security scheme should be usable for people of limited technical background.

Inclusive. Since not all patients own smartphones that run sophisticated mobile apps and support security features, the system should be available to patients with only SMS and phone call capabilities.

3.1 Threat Model

Before describing the system design, we clarify a few assumptions about the adversaries. First, we assume an attacker can spoof the healthcare provider's phone number when sending SMS to patients. The attacker's goal is to send a high volume of phishing messages to many users, expecting some of them to perform the intended actions (e.g., revealing passwords or credit card information). These adversaries represent those that run small mobile carriers or VoIP services.

Second, for strong adversaries, we also assume they are able to spoof the patient's phone number to contact the healthcare provider (e.g., attempting to extract patient information). While this attack capability is not necessarily a major concern given healthcare providers can order premium services from mobile carriers that offer anti-spoofing and call filtering [84], we will still discuss how we mitigate this threat in our design.

Third, we assume the basic integrity of the devices owned by the patients and healthcare providers, e.g., attackers did not infect/control the patients/hospital devices with malware.

Finally, we assume attackers cannot actively intercept and tamper with the communication between patients and the healthcare providers as "man-in-the-middle" (or MITM). This capability is unrealistic for adversaries in this context given the high attack cost (e.g., setting up fake base stations physically approximate to the victim). The cost is unlikely worthwhile given the low profit-conversion rate of such campaigns [30].

4 SYSTEM DESIGN

The goal of VeriSMS is to significantly increase the efforts/costs for an attacker to spoof a healthcare provider when sending phishing messages to patients. In this section, we describe the design of VeriSMS and discuss why alternative designs are not considered. Then, we perform a security analysis to discuss the tradeoff between security and usability.

4.1 A Call-To-Verify System

Fig. 1 describes the high-level idea of VeriSMS, which is a call-to-verify system. When users receive a message that appears to be sent by their healthcare provider, they can make a phone call to verify the authenticity of the message. The healthcare provider will host a programmable *voice agent* behind its official phone number to provide the verification service.

As shown in Fig. 1, the message carries a *Message ID*, which contains random numbers, and *Secret Words*, which are English words of everyday objects. To verify the message, the user can call the healthcare provider (based on the phone number on their hospital card) and enter the Message ID. After verifying the message ID is valid, the voice agent will read back the two Secret Words. If both Secret Words match with those in the message, then the message can be determined as "authentic." Otherwise, the message cannot be trusted. Note that Secret Words are uniquely associated with each patient (based on the patient's phone number).

Note that, if the Message ID is determined as "*invalid*" by the voice agent, the message is also regarded as untrusted. In this case, the voice agent will not proceed to the next step (i.e., it will not read back the Secret Words). The purpose is to prevent attackers

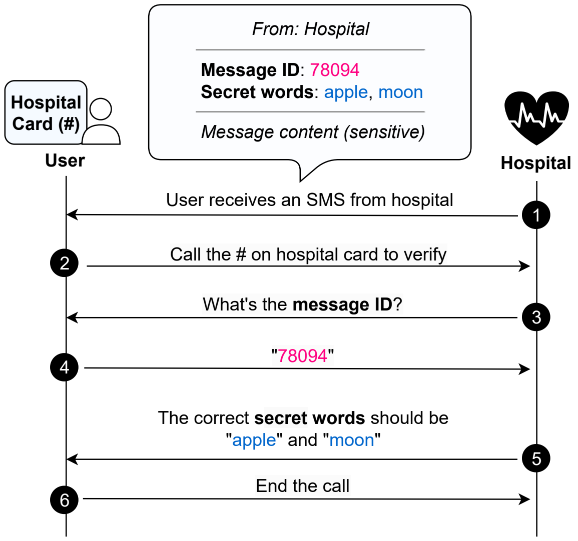

From: Hospital
Message ID: 78094
Secret words: apple, moon
Message content (sensitive)
Hospital Card (#)
User
Hospital
User receives an SMS from hospital
1
Call the # on hospital card to verify
2
What's the message ID?
3
"78094"
4
The correct secret words should be "apple" and "moon"
5
End the call
6

Figure 1: VeriSMS Workflow—The workflow to verify a message by a user who receives a message from the healthcare provider.

from spoofing the user's phone number to interact with the voice agent to extract the user's Secret Words. We will further discuss this threat Section 4.3.

A key advantage of this design is that it requires no additional hardware from users. To minimize the cost of enrollment, we piggyback on an existing mechanism of healthcare providers, which is the physical cards/papers that patients obtain from the healthcare provider (e.g., during their in-person visit). This physical card will serve as the "*root of trust*" as it carries the healthcare provider's true phone number. This ensures the patient will call the correct number to verify the message.

4.2 Verification Scheme Design

VeriSMS has gone through several design iterations. In the following, we discuss a few naive baselines and explain why we do not use them. This will help to better understand our final design choices.

4.2.1 Why not Static Code Design. To verify a message, one naive design is to use a *static code*. The example message is shown in Fig. 2a. The healthcare provider can generate a static code for each patient. Each patient receives a unique code which is generated based on the patient's phone number. For a given user, *the same static code* will be attached to all the messages sent by the healthcare provider. Users do not need to call the healthcare provider to verify the code. It requires an initial code setup (e.g., in-person), and then the users can memorize the same code, expecting it to appear on all messages from the healthcare provider.

This scheme is similar to conventional mnemonic images [9] or Sitekey [72] that have been used for web login/authentication. The main concern about this design is security. Since the same code is used for all the messages to a user, code leakage (via shoulder surfing, code guessing) will compromise the scheme. In other words, if the adversary used the correct static code, the phishing message would be fundamentally indistinguishable from a benign one. In addition, setting up the code or changing the code may not be feasible remotely (e.g., needs to be done in person) given the lack of a trusted communication channel in the first place.

4.2.2 Why not One-Time-Code Design. Another alternative design is to prioritize security by using one-time code, as shown in Fig. 2b. In this design, the healthcare provider would generate a pair of random numbers to act as the "Message ID" and the "Secret", based on the patient's phone number. For each new message sent to this patient, a fresh pair of Message ID and Secret will be generated for the message. Since the Message ID and Secret are randomly generated each time (i.e., not re-used), it helps to address the code leakage concern in the previous static-code design. To verify the message, the patient will need to call the healthcare provider, enter the Message ID, and check if the returned Secret matches with the one on the message.

While the one-time code has the advantage of providing *high security*, our partners from healthcare provider OSF Healthcare have expressed their concerns about *usability*. One-time code requires users to call the healthcare provider *every time* whenever they receive a new message—the conjecture is that users are unlikely to do so in practice. In addition, the Secret provides little semantic information for users, which can be challenging for users to understand its meaning.

4.2.3 VeriSMS: Hybrid Design. Based on the feedback from our industrial partners, we use a hybrid design for VeriSMS. Instead of aiming *exclusively for strong security*, we adjust our goal to significantly increase attackers' costs while prioritizing practicability and usability. As shown in Fig. 2c, we still use a pair of Message ID and Secret. The Message ID is a randomly generated number changed each time (unique for each new message), but the Secret is *static* (unique for each user/phone number). If needed, users can still change the Secret remotely and on demand by calling the voice agent. This hybrid design seeks to preserve the capability of detecting caller ID spoofing as well as resilience against secret leakage.

For each message, the Message ID[2] is a 5-digit random number, and the "Secret Words" are two English words remaining static. Both data fields are uniquely associated with a patient's phone number. In practice, the Message ID may use more digits to increase the guessing difficulty—the corresponding cost is that honest users will need to enter a longer ID during message verification. For the Secret Words, we use *two* English words instead of one, to significantly increase the difficulty of adversarial guessing. For example, given a pool of 4,000 words, there are 16,000,000 possible two-word combinations. As mentioned in the threat model (Section 3.1), we primarily consider attackers who aim to maximize their profits by sending messages to a large number of users. The two-word combination (unique to each phone number) will significantly increase the costs of such campaigns.

Compared with the *one-time code* design above, this hybrid design sacrifices some level of security as the Secret Words are static (unless users request to change them). The main benefit in return

[2] In our initial design, we named this Message ID as "hint code". During our exploratory user study, we learned that the name "hint code" can cause confusion to users. As such, we changed the name to "Message ID" in the final design. See the detailed discussion in Section 6.

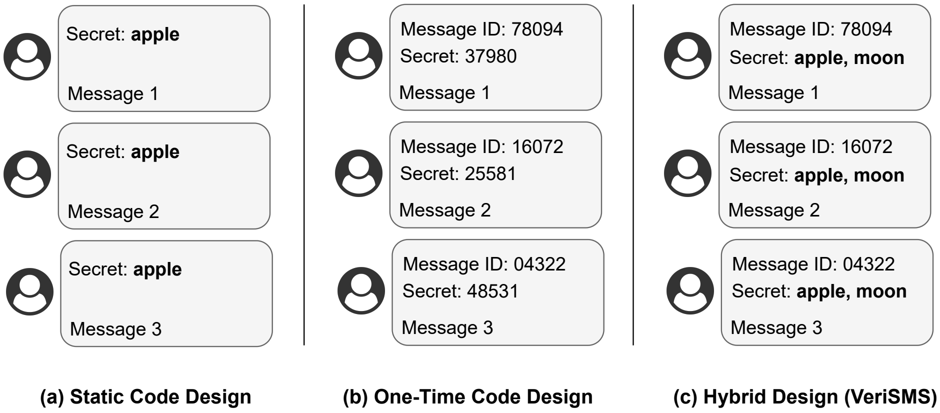


Figure 2: Example Messages—VeriSMS adopts a hybrid design to balance security and usability.

is the improved *usability*—users can quickly glance at the Secret Words on top of the message to identify phishing messages (if they have the wrong Secrets). In this way, users do not need to call the voice agent for every message they receive. Compared with the *static code* design, this hybrid design has improved security because it uses a one-time random Message ID and allows users to call the voice agent to verify the message (i.e., even if the message has the correct Secret Words, it is unlikely to have the correct Message ID). This gives the users the option to verify the message, especially when (1) the users find the content of the message suspicious, and/or (2) the message contains sensitive/important instructions (e.g., payments). It preserves the resilience to secret leakage as long as the users call the voice agent on important messages.

4.3 Security Analysis

We perform a security analysis to understand (1) how VeriSMS raises the attackers' costs under different scenarios, and (2) how it stays resilient against adaptive attacks (i.e., attackers that become aware of VeriSMS and make adaptations).

Users Always Call to Verify Important Messages. In the ideal scenario, users always call the voice agent to verify an important message. Important messages are those that instruct users to perform sensitive actions such as clicking on a link or making a payment. In this setting, to deceive users, an adaptive attacker will need to correctly guess the Message ID as well as the two-word combination *for each user* and *for each message* sent. Suppose there are 4,000 words in the Secret Word pool and VeriSMS uses a 5-digit Message ID, the chance of a correct guess is one out of 100,000 × 16,000,000, which is $1.6{\times}10^{-12}$. As a reference point, prior work on the spam ecosystem [30] shows that the conversion rate from email spamming to sales is less than 0.00001%. It is a reasonable expectation that VeriSMS can significantly raise the barrier of performing large-scale SMS-based spam/phishing campaigns.

Users Never Call to Verify Important Messages. This is likely the worst-case scenario. Without calling the voice agent, users will only rely on the memorable Secret Words to infer the authenticity of the message. In this setting, to deceive users, an adaptive attacker must correctly guess this user's secret word combination in one shot (or in a few tries). Otherwise, the user would soon realize the anomaly given the inconsistent Secret Words. In this case, suppose the attacker managed to obtain a user's Secret Words through a side channel, then VeriSMS is not effective (equivalent to the static code design). However, if the attacker tries to guess the secret word combination, the chance of success is still low. For a pool of 4,000 words, the chance of success is one out of 16,000,000, which is $1.6{\times}10^{-7}$. Again, this is the worst-case scenario—users can protect themselves as long as they call to verify *important* messages.

Attackers Calling the Voice Agent by Impersonating a User. One adaptive attacker may spoof a target user's phone number to call the voice agent, aiming to obtain the user's Secret Words. In this case, the attacker will need to first guess the Message ID, with a success rate of only 1×10^{-5}. As mentioned in Section 4.1, when the Message ID is wrong, the voice agent can choose to randomly return the wrong Secret Words instead of telling the caller that the Message ID is wrong. In this case, the attacker (who impersonates the user) can no longer tell if their guess is correct or not. In addition, healthcare providers can set a *threshold* for the number of allowed verification calls per day per phone number. The intuition is that one honest patient is highly unlikely to call more than a few times a day to verify the message sent from the system.

Omission Attack. An attacker may choose not to include any Message ID or Secret Words in the message. When a user suddenly receives such a message (without any Message ID or Secret) from the healthcare provider's number, the correct reaction is to treat it as "*untrusted*". However, in practice, we suspect users may get confused by such messages. As such, we will examine this adaptive attack in our user study experiments.

Other Adaptive Attacks. We acknowledge the above discussion cannot exhaustively cover all possible adaptive attacks. More complicated adaptive attacks (i.e., those that aim to mislead users on the root-of-trust) will be further discussed in Section 8.

4.4 Prototype

We developed a prototype for VeriSMS, which will be used for the later user study. The prototype consists of an SMS message sender and a programmable voice agent—both are running on a server that hosts a database to store user information. We use a VoIP service Twilio [79] to implement the SMS sender and the voice agent. For the database, we use SQLite for storing data fields such as a user's phone number, and each message's Message ID and Secret Words.

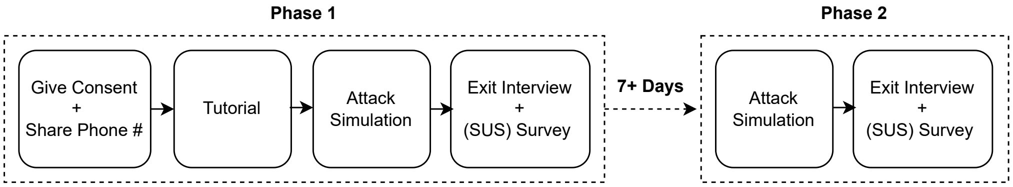

Phase 1
Give Consent + Share Phone #
Tutorial
Attack Simulation
Exit Interview + (SUS) Survey
7+ Days
Phase 2
Attack Simulation
Exit Interview + (SUS) Survey

Figure 3: User Study Workflow—The study contains two phases with a 7-day gap in between. During phase-1, the participant uses their personal mobile phone to do a short tutorial, followed by an attack simulation, and an exit interview/survey. During phase-2, the participant will not go through any tutorial and directly start with the attack simulation, followed by an exit interview/survey.

We purchased a static VoIP phone number as the number of the healthcare provider (which is also the number for the voice agent). Fig. 8 (in the Appendix) illustrates the voice agent's workflow when interacting with incoming callers.

5 USER STUDY METHODOLOGY

We conduct a user study to explore potential problems that VeriSMS may face during deployment, revise the system design based on user feedback, and further validate these changes. More specifically, the user study is first run with N=15 participants for exploratory purposes (*exploratory study*). This run returns valuable lessons to revise the system design. After revision, we ran the study with another group of N=35 participants for validation purposes (*validation study*). The two studies follow a similar procedure. In the following, we first describe this general procedure and then detail the differences between the two studies. The user study seeks to answer the following research questions.

- **RQ1.** How well can users understand and use the system correctly?
- **RQ2.** How well can users use the system to identify spoofed phishing messages? What are the reasons behind potential false positives and false negatives?
- **RQ3.** How usable is the system to users? What factors have affected the perceived usability?

5.1 Study Workflow from Participants' View

We start by describing the user study procedure from a participant's perspective using Fig. 3. The study is conducted online over a video conference call (using Zoom) and it contains two phases.

Phase-1. A participant starts by reading the consent form and giving their consent. Then they share their personal phone number with the researcher. In this study, each participant uses their personal phone (which has the participant's familiar messaging interface) to receive and browse text messages in order to create an authentic experience. The participant is reminded that the phone number will only be used during the study process, and will be deleted by the research team after the study. The participant is also informed that the session is audio-recorded.

Then the participant will go through a short tutorial. The participant is told to imagine a scenario of visiting their (fictional) doctor's office and learning that a new patient outreach system has been introduced. They are then told to read a short instruction about the system. To do so, the researcher shares a URL with the participant over the Zoom chat. By clicking on the URL, the participant will view a single web page with written instructions about VeriSMS. This instruction explains the verification process step-by-step and teaches how to determine whether a message is authentic. In practice, we envision that this tutorial can be given to users via printed handout during their visit to the doctor's office. Along with the written instructions, we also present a hospital card (on the web page) that carries the phone number of the hospital. The participant is explicitly told they can trust the instructions and the hospital card. While reading the tutorial page, the participant will receive the first welcome message from the hospital which contains a Message ID and their Secret Words. The participant can ask any questions about the system. They are also instructed to test the voice agent by calling the phone number on the hospital card, to verify the welcome message.

With the participant confirming that they have understood how the system works ends the tutorial stage, the study then progress to the attack simulation stage. During the attack simulation, the researcher will no longer answer any questions from the participants (nor ask them questions) to minimize priming, until the end of this step. The participant will receive both benign and spoofed messages, one at a time (7 messages in total). For each message, the participant has an unlimited amount of time to read and assess the message. They still have access to both the instruction document and the hospital card. The participant can choose either to call the voice agent to verify the message or make their determination without calling. After their assessment, they will inform the researcher whether they believe the message is an authentic message from the hospital or a fraudulent one. Further details about the attack simulation are presented in Section 5.2.

In the final step of phase-1, the participant will do a brief exit interview and a survey. The participant is asked about their decision-making on each message during the attack simulation. Then they will complete a usability questionnaire and answer demographic questions. This step is further detailed in Section 5.3.

Phase-2. Phase-1 performs attack simulation right after the tutorial session. A potential concern is that users may be able to use VeriSMS correctly *right after* the tutorial but it is unclear how well this knowledge can be preserved over time. As such, after phase-1, we wait for at least 7 days to invite (some of) the participants back to do a second study to re-assess their ability and willingness to use VeriSMS (under the possibility that users might forget some of the details in the instruction/tutorial). As shown in Fig. 3, phase-2 no longer has the consent step (the consent form completed in phase-1 covers both phases) or the tutorial step. Instead, the participant starts directly from the attack simulation.

Message Type	# of Messages	Configuration
Benign	4	Correct Message ID and correct Secret Words
Fraud-A	2	Incorrect Message ID and incorrect Secret Words
Fraud-B	1	A plain message without Message ID or Secret Words

Table 1: Messages in Attack Simulation—Each participant will receive 7 messages on their personal mobile phone in a randomized order during the attack simulation.

5.2 Messages for Attack Simulation

We include three different types of messages for attack simulation. For each participant, in a given attack simulation session, they will receive 7 messages on their personal mobile phone, in a *randomized order* (Table 1).

- **Benign.** 4 messages are benign messages that contain the correct Message ID and the correct Secret Words.
- **Fraud-A.** 2 messages are spoofed messages that contain a wrong Message ID that cannot be recognized by the voice agent and the wrong Secret Words.
- **Fraud-B.** 1 message is a plain spoofed message that does not contain a Message ID or Secret Words and only has the text content. This is to simulate an *adaptive omission attack* described in Section 4.3 where adversaries intentionally omit such information in spoofed messages.

To make the study manageable and reduce the workload of participants, we did not simulate other (adaptive) attacks. There are other potential attacks such as sending messages with the correct Message ID and one/two wrong Secret Words, or sending messages with the wrong Message ID and correct Secret Words. Part of the reason for not including them is also the low probability of guessing the Message ID or Secret Words (see Section 4.3).

We draw the text message content from a pool of messages that are actually used during real-world patient outreach by the healthcare provider OSF Healthcare. The message content covers various patient outreach needs such as appointment reminders and notification of payment dues. We provide all the messages used in the study in the supplementary materials [3]. To reduce biases from specific message content, we *randomly* assign the message content to the 7 of the message configurations listed in Table 1. This means, a given message content can be used by a *fraud* message if the Message ID/Secret Words are wrong, and the same content can be used by a *benign* message when the Message ID and Secret Words are correct. This reduces the biases introduced by the content itself. The rationale is that real-world attackers can send messages with authentic-looking content (with a malicious URL). To protect participants, we replaced all existing URLs in the message with an *unclickable* placeholder "`{LINK_HERE}`."

Design Considerations. There are two important design considerations worth further discussion. First, our attack simulation did not include a comparison group where *VeriSMS is not used.* This is because (1) abundant evidence from prior research and reports from the Federal Trade Commission (FTC) [4, 21, 31, 41] has already shown that users, without additional protection, are highly vulnerable to SMS-based phishing. The purpose of our study, instead, is to show whether users can understand VeriSMS correctly and use it to detect phishing messages. (2) For a baseline without *VeriSMS*, users can only rely on the *message content* and *caller ID* for phishing detection. However, as discussed above, attackers can use *exactly the same wording* of the authentic message (except for using a malicious URL) and spoof the hospital's phone number. In other words, neither message content nor caller ID are reliable features for phishing detection, and thus it is not needed to test users under such situations.

Second, while our experimental attack is a targeted attack (i.e., impersonating the victim user's healthcare provider), we did not take extreme measures to customize the message by including the target user's name or pronoun. The reason is to *stay consistent* with OSF Healthcare's patient outreach message content which does not include the patient's name. We note that existing work shows that spear phishing messages that reference the target user's name or other personal information can make the message even more deceptive [41]. We leave the investigation of more targeted attacks to future work.

5.3 Exit Interview and Survey

After the attack simulation, we perform a brief interview with participants, going through *each of the messages* they received and asking the following questions:

(1) How confident are you about your judgment of this message? (5-Likert-Scale) (2) Why do you choose to call or not call the verification system for this message? (3) After calling the verification system, did you find the result unexpected? Did the call change your initial judgment of the message? (4) The system does allow you to change your Secret Words—under what condition would you like to change your Secret Words?

After answering the questions, the participant is then instructed to take a survey to answer questions about the system's usability, and demographics (age, gender, and educational background). For the usability survey, we take the standard System Usability Scale (SUS) [6, 80]. This usability survey contains 10 questions, and each question provides opinions on a 5-point Likert scale from "Strongly Agree" to "Strongly Disagree." These questions assess various usability aspects such as "*I found the system unnecessarily complex*" and "*I think that I would like to use this system frequently.*" We will aggregate the SUS score to assess the overall usability of the system. We provide the complete question list (as well as other user study materials such as the tutorial web page) in the supplementary materials [3].

5.4 Recruitment and Ethics

Our study was reviewed and approved by our Institutional Review Boards (IRB). We recruited participants from the Prolific platform between January and November 2023. We did not collect personally identifiable information (PII) (other than the phone number, which is only used during the study sessions and is deleted afterward). Participants can withdraw their data at any time after completing

	Explor.	Valid.
Number of Participants	**15**	**35**
Gender:		
Male	8	15
Female	7	20
Other	0	0
Age:		
18-24	1	2
25-34	5	7
35-44	3	9
45-54	1	2
55-64	3	11
65+	2	4
Education:		
High School Graduate or Less	1	3
Some College/2-year Associate	2	15
Bachelor's Degree	5	12
Some Graduate School	2	0
Master's or Professional Degree	4	4
Doctoral Degree	1	1

Table 2: Demographics—the demographics information for exploratory study (Explor.) and validation study (Valid.), respectively.

the study. We recruited participants from the United States, (1) to match the patients' demographics of the healthcare provider OSF Healthcare, and (2) to accommodate the time zone constraints of running the study. Each participant would receive $8 if they participated in phase-1 and would receive an extra $12 if they also participated in phase-2. On average, phase-1 takes about 25 minutes and phase-2 takes about 20 minutes.

The demographics of our participants (50 in total) are summarized in Table 2. First, we recruited N=15 participants for the *exploratory study*, and we used the collected feedback and results to revise the system design. Then, we recruited a different group of N=35 participants for the *validation study*. While *VeriSMS* is designed for all users, we particularly want to make sure it works well for older adults as discussed in Section 1. As such, we intentionally over-sampled older adults for the validation study. Out of the 35 participants, 15 (43%) are over 55 years old. All 35 of them participated in phase-1, and a random subset of 9 participants were invited back to participate in phase-2 to confirm they can still correctly use the system after some time gap (Section 5.1).

5.5 Data Analysis Method

There are three primary types of data collected in the study: (1) message classification results (i.e., for each message, we record the participant's answer about whether the message is "fraudulent" or "benign"), (2) SUS scores collected during the exit survey, and (3) participants' responses recorded during the interview session. Based on the data, we focus on the following analysis.

First, to understand participants' message classification performance, for *each participant*, we calculate the number of fraud messages correctly identified (out of 3) and the number of benign messages correctly identified (out of 4). This is equivalent to a true positive rate (TPR), and a false positive rate (FPR)[3]. We report the number instead of the rate, considering the small number of messages that each participant reviewed.

Second, to evaluate the perceived usability, we calculate the overall SUS score. Recall that SUS has 10 standard usability questions [6, 80], and we follow a common approach to calculate the overall SUS score for each participant. More specifically, each question's 5-point answer is first mapped to a score ranging from 0 to 10. Then, a participant's score is the sum of their ten responses (i.e., the total ranges from 0 to 100).

Finally, to understand the behavior of (and concerns from) participants, we transcribed the interview recordings and analyzed the data using thematic analysis [5]. Two coders first each coded five different interview sessions independently, and then discussed each other's codes, iterated upon the codes, and finalized a draft codebook down to fine granularity. Both coders then independently coded four more common interview sessions to verify and finalize the codebook. The codebook has 39 codes, categorized into five high-level themes: "Fraud Detection", "Usability Concerns", "Strategies for Managing Secrets", "Strategies for Calling", and "Suggestions for Improvements." We have a third coder who coded a subset of six interview sessions using the finalized codebook independently, to verify the inter-coder reliability of the codebook. We calculated the inter-coder reliability based on the subset of 6 interview sessions with a percentage of agreement of 89.5% and 0.884 in Cohen's Kappa, which is considered substantial agreement. The remaining study sessions were coded using this finalized codebook.

6 RESULTS: EXPLORATORY STUDY

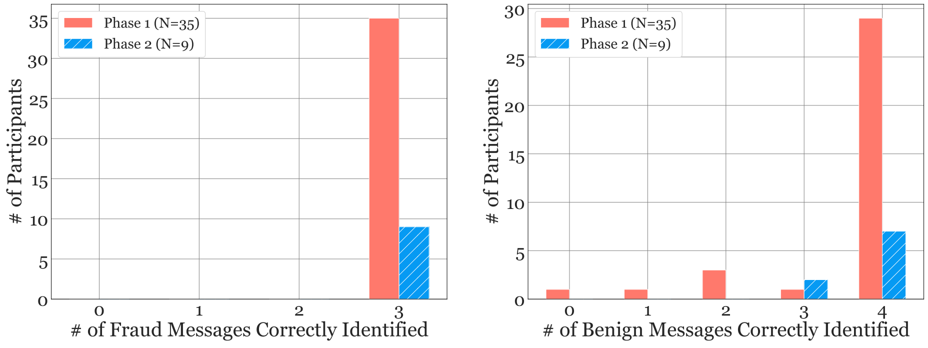
We first briefly discuss the results of the exploratory study (N=15). This section will focus on *the lessons learned from this exploratory study* and how we use the results to revise the system designs and the user study procedure.

Phone Verification Workflow. A key feedback from participants is to improve the VeriSMS verification workflow by *reading back* the Message ID entered by the user. Without this *reading-back* step, participants may accidentally enter a digit of the Message ID wrong, causing failed verification. In the revised version, the voice agent will read back the entered Message ID, and ask for the user's confirmation before moving to the next step.

Message ID. In the initial design, the Message ID was named "hint code," which has caused some confusion in this exploratory study. Some participants asked why the hint code is different for every message. Based on this observation, we revised the design and renamed it as "Message ID." As an "ID" of the message, by definition, it is expected to be distinct for each message. We also clarified the meaning of the Message ID in the tutorial instruction to reduce potential confusion among participants.

Clarifications in Tutorial. In the original user study design, the tutorial page displays an example message screenshot to explain what Message ID and Secret Words are. However, in the exploratory study, some participants mistakenly regarded the example Secret Words as the real ones associated with their phones. In the revised version, we clarified by stating that the one on the tutorial page was just an example, and we also explicitly added the keyword "example" to it (i.e., "`example_word_1 example_word_2`").

[3] More precisely, it is equivalent to 1− false positive rate.



(a) Detected Fraud Messages

(b) Detected Benign Messages

Figure 4: Message Detection Results—Correctness of message classification by participants in the validation study.

URL in Messages. In the initial user study design, not all messages contain URLs—this is because certain outreach messages used by healthcare provider OSF Healthcare do not contain URLs in the first place. During the exploratory study, some participants indicated that the presence of URLs influenced their assessment. To reduce such influence, we added the same URL placeholder "`{LINK_HERE}`" to *all messages* in the revised study.

7 RESULTS: VALIDATION STUDY

In this section, we focus on the validation study to examine the effectiveness, usability, and potential deployment issues of the proposed system. Given the revisions made to the system and user studies described above, we still will report the result from the exploratory study but restrain ourselves from formally comparing the two studies' results. The result from the exploratory study will be presented in Appendix A as a reference, to confirm the aforementioned issues in Section 6 have been resolved by the revision.

7.1 Spoofing/Fraud Detection Performance

For each message, the participant will determine whether it is *benign* or *fraudulent*. This determination is made by reading the message content, checking the Message ID and Secret Words, and/or calling the VeriSMS voice agent.

Overall Classification Performance. Fig. 4 shows the classification results on benign and fraudulent messages, respectively, for the validation study. Recall that we recruited in total of N=35 participants in the validation study for phase-1 and 9 participants were randomly selected to be invited to participate in phase-2.

Fig. 4a shows that *all* participants have successfully identified *all* fraudulent messages, during both phases. Both Fraud-A messages and Fraud-B messages are correctly identified. Recall that fraud-B messages are an adaptive attack where adversaries intentionally omit the Message ID and Secret Words to prevent message verification. All participants have successfully determined such messages as "fraudulent." The high detection rate indicates that participants can correctly understand the system to perform spoof detection (RQ1).

Fig. 4b shows that the vast majority of participants have correctly identified all benign messages during phase-1 (29/35, 83%), and phase-2 (7/9, 77%). Most participants during phase-1 (30/35, 86%) and all participants during phase-2 (9/9, 100%) have identified three out of four benign messages. Only a few participants have made incorrect determinations on certain benign messages (despite that they carry the correct Message ID and Secret Words). The reasons behind the false positives will be further discussed in Section 7.2 via a qualitative analysis of the exit interview.

It is also worth mentioning that older participants (age of 55+) had comparable performance with the rest of the group. On one hand, all of these 15 older participants have successfully identified all fraudulent messages. On the other hand, 11 of them (73%) correctly identified all benign messages.

Overall, the result answers **RQ1** that participants can correctly understand and use VeriSMS. It also partially answers **RQ2** that participants are able to achieve a high detection accuracy using VeriSMS. While the good performance might be related to the priming effect of the *tutorial session* right before the attack simulation, we mitigate this concern with phase-2, as we run the study again after a time gap (without the tutorial session).

Phase-1 vs. Phase-2. There is a gap of at least 7 days between phase-2 and phase-1 to examine whether participants can still correctly use VeriSMS after receiving the tutorial for some time. Overall, 83% of the participants in phase-1 correctly classified all 7 test messages and 77% participated in phase-2 correctly classified all 7 test messages. The overall result suggests that most participants still remembered how to use VeriSMS after 7+ days.

We further examine the two participants who misclassified 2 (out of 4) benign messages during phase-2 (see Fig. 4b). One participant made the determination based on the language and grammar of the text messages, despite the fact that the Message ID and Secret Words are correct. The other participant indeed forgot certain details of VeriSMS: "*I forgot the fact that the Message ID will be different for each message compared to my previous messages...*" This participant realized/recalled this detail when making a call to the voice agent to assess the second benign message. After calling, this participant proceeded to correctly classify all the rest of the messages. Overall, the result indicates that most participants can remember how to use VeriSMS after some time gap. In addition, calling the voice agent seems to be an opportunity to remind users about how to use VeriSMS.

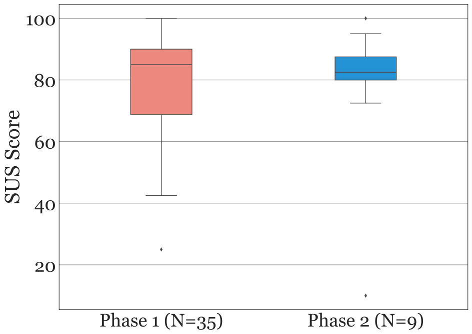


Figure 5: Usability Results — The SUS score of each participant in the validation study.

7.2 Usability

To examine usability, we analyze the responses collected during the exit survey and interview. This Section will focus on analyzing the descriptive statistics, and Section 7.3 will present the qualitative results.

Usability Scores. Fig. 5 summarizes the participants' responses to the system usability scale (SUS) questionnaire. According to [6, 80], the usability of the system is "above average to good" if the mean SUS score is higher than 68. The system's usability is considered "excellent" with a score over 80.

As shown in Fig. 5, during phase-1, we received a mean SUS score of 79.1 (with a median of 85). This score is between "good" and "excellent." The mean score is lowered mainly by a few participants giving low ratings, and their reasoning will be further discussed below with the qualitative analysis. To assess the perception of older adults, we calculate the mean SUS score for participants older than 55 years old, which is 79.3 (with a median of 77.5). This shows the aging participants accept the design of VeriSMS.

For phase-2, the mean SUS score is 77.5 (with a median value of 82.5). The score is again between "good" and "excellent." Overall, the result answers **RQ3** that the usability of the system is perceived positively by most of the participants.

Phase-1 vs. Phase-2. Comparing phase-1 and phase-2, the perceived usability (after 7+ days) remains at a high level. Also, the variance of the usability score becomes even smaller. A possible explanation is that participants get more familiar with the system as they use it more. That being said, we still observe the low SUS score for one participant (P1), who is the *same participant* giving the lowest SUS score during phase-1. The participant believed that message verification was not necessary since they were capable of identifying the phishing message by reading the message content alone: "*I can usually just eyeball it to identify it*" [P1, F, 18-24][4].

7.3 Practical Issues Related to Deployment

To understand the behavior of (and concerns from) participants, we have performed a qualitative analysis using the method described in Section 5.5. We discuss the key findings below.

Reasons Behind False Positives. Analyzing the interview results on the few false positive cases (see Fig. 4b), we identify three main reasons. First, participants made their decisions based on the *message content*, despite the correct Message ID and Secret Words: "*I think those are suspicious because of the content*" [P2, F, 55-64].

Second, participants accidentally made a mistake when entering the Message ID to the voice agent and thus the benign message failed the verification. For example, one of the benign messages has a Message ID of "72712" and participant [P3, M, 25-34] entered "72717" into the system (which was off by one digit).

Third, participants got confused after they changed/reset their Secret Words. More specifically, [P4, M, 25-34] chose to change their Secret Words via the voice agent during the attack simulation. Shortly after, they got a notification stating that their Secret Words had changed. However, as this participant proceeded with the next message, they made a mistake (false positive) due to the new/unfamiliar Secret Words. This participant realized the mistake afterward and asked to revise their answer—we still documented this as a false positive based on their first decision. This indicates a need to further improve the confirmation/notification of Secret reset and remind users about the new Secret Words.

Reasons Behind Low SUS Scores. To further answer **RQ3**, we examine the reasons behind the few low SUS scores during the exit survey (see the outliers in Fig. 5). The low score is from participants who don't (want to) use SMS in real life when communicating with healthcare providers. For example, [P5, M, 55-64] stated: "*If this is real life, I would not want alert and text messages from doctors. I have an eye doctor do this and I cannot withstand it ... But, if you ask me about the system, in case it's payment-related and after calling the system a few times, I think the system itself is a good system. This is more secure than plain messages.*"

Another reason comes from a complication from using VeriSMS, which suggests room for improvement. More specifically, when a user calls the voice agent, the voice agent would ask for the Message ID. In order to enter the Message ID, the user will need to look it up by switching the screen back to their SMS application. Then, they memorize the Message ID, switch back to the phone call, and enter each digit. This process is inconvenient for certain participants. "*When you are on your phone, and you have to switch screens from text to phone call and remember the number, it's not that hard but very tedious*" [P6, F, 35-44].

Do People Call Voice Agent? Why (Not)? Whether and when users call the voice agent would influence the security level of VeriSMS (see Section 4.3). Our qualitative analysis reveals users' strategies for such decisions. Given that not all participants had phase-2, when reporting descriptive statistics below, we only consider phase-1.

8 out of 35 participants (23%) stated that they would always call. The reason is to, for example, ensure the validity of their current Secret Words: "*I will always call and I also want to check if my secret word is never changed*" [P7, M, 35-44].

13 out of 35 participants (37%) stated that they would call selectively, under certain conditions. The most common calling situation is when the message content looks suspicious or the message contains important information. "*In real life, yes I am gonna verify but not as often. I may call if (the message about) appointment or payment*

[4] When representing quotes from participants, we will mark their gender and age group. For instance, [P1, F, 18-24] refers to participant number 1 who is a female and from the age group of 18–24 years old.

is in doubt; if I recognize it (Secret Word), I would not have to verify it" [P8, F, 55-64]. The other stated calling situation is when the presented secret words are incorrect, and the participants would call to confirm whether the message is indeed fraudulent. "*Most of the time I would not call. I'm fine with the secret code and non-damaging reminder. If it's asking for money with the wrong secret I ignore it*" [P5, M, 55-64].

13 of 35 participants (37%) stated that they would never call and would only rely on the correctness of Secret Words to determine whether a message is trusted. The reason is that participants believed that the chance of attackers guessing their correct Secret Words was low. "*It's hard enough to crack the secret word*" [P9, F, 25-34]. Our security analysis (Section 4.3) confirms the high difficulty of guessing a target user's Secret Words (given adversaries do not have the opportunity to make a large number of guessing attempts). However, users can be vulnerable if the Secret Words are leaked in a rare event (e.g., via a side channel).

Finally, one participant (2%) stated that they would never call because they would not use SMS at all (which is likely not a target user for VeriSMS).

User-suggested Design Improvements. Finally, we summarize user suggestions to further improve VeriSMS for potential deployment. First, participants suggested some easy-to-implement improvements such as raising the default volume level of the voice agent: "*make it a bit louder*" [P10, M, 35-44], increasing the time allowed to enter the Message ID: "*when I am entering (Message ID), it didn't allow enough time*" [P11, F, 55-64], and using more concise languages for the voice agent "*It would be better if they (voice agent) just said your secret is XYZ*" [P12, F, 35-44].

In addition, a few participants suggested that the voice agent should "*explicitly confirm the (Message) ID is matched*" [P13, M, 18-24], before it proceeds to read the Secret Words. As described in Section 4.2, the voice agent may choose not to provide the explicit confirmation about the correctness of the MessageID, and then (1) provide the correct Secret Words when the message ID is correct or (2) provide the incorrect (random) Secret Words when the message ID is incorrect. This prevents attackers from knowing whether their guessing is correct. However, the user study result shows that the lack of explicit confirmation also hurts normal users' experience. As such, in practice, the voice agent may provide explicit confirmation and use other ways to prevent attackers from brute-forced guessing (e.g., setting a daily limit on the number of verification calls per phone number).

8 DISCUSSION

Security and Usability Trade-off. The user study results confirm our initial concern about exclusively going after strong security. More specifically, a more secure version of VeriSMS would be using random codes for both Message ID and Secrets for each message (see the one-time code design, Fig. 2b). However, the first challenge would be effectively explaining how this system works to users (if both are random codes). As observed in the Exploratory Study (Section 6), participants need to understand the meaning of the code. In the initial version of VeriSMS (used in the exploratory Study), the Message ID was named "hint code," which affected users' ability to interpret its meanings (i.e., why it is different every time). In addition, the more secure version depends on the assumption that *users would call every time*. Otherwise, it would *lose all the security benefits*. In comparison, the current design (i.e., using static English words as the secret) would still preserve a reasonably high security level even if users don't call or only call on sensitive/important messages. As shown in Section 7.3, the majority of the participants do not call on every message. Overall, VeriSMS prioritizes the usability/practicality aspect, while achieving the goal of significantly increasing the attacker's costs.

VeriSMS vs. Prior Works and Existing Solutions. In the following, we further discuss how our work is related to and different from existing solutions. First, the idea of Secret Words may appear similar to SiteKey [72, 87] (or Security Images) originally used in web authentication scenarios. SiteKey was once used by websites such as Bank of America and The Vanguard Group [87, 91]. However, there are fundamental differences between SiteKey and VeriSMS, and VeriSMS also addresses a well-known vulnerability in SiteKey. SiteKey is for web authentication: when a user tries to log in to a bank website, once the user enters the user name (e.g., the email address), the website will display a secure image (e.g., a cat image, as the site key) to the user. The secure image is pre-selected by the user when this user sets up the bank account—if a wrong image is presented, it indicates the website is not the real one (i.e., a phishing site), and the user should not continue to enter the password. A well-known vulnerability of SiteKey [91] is any attackers can impersonate the user to get the user's secure image: the attacker first visits the bank website to enter the user's email address, and the bank will display this user's personal secure image (based on the email address). Then this attacker can create a phishing website (mimicking the bank site) to display the correct secure image to users. As such, SiteKey is fundamentally flawed. In our case, VeriSMS does not have this vulnerability because (1) the message (that contains Secret Words) is sent by healthcare providers directly to the user's phone, which is not visible to the attacker; (2) if the attacker impersonates the user to call the voice agent, attempting to extract the Secret Words, they will also fail given they don't have the correct Message ID. Overall, VeriSMS adapts the idea for message verification and has proactively removed known vulnerabilities.

Compared with existing phishing studies focused on phishing websites [1, 29, 34, 76] and phishing emails [13, 55, 85, 88], we take a different approach. More specifically, phishing emails and phishing websites have a rich set of cues (e.g., URLs, phishing website layout, input box, email content, email sender) that users rely on to perform phishing detection [1, 76, 85, 88]. Prior studies have investigated how different cues affect users from different demographic groups [4]. However, for SMS, the phishing cues are much more limited (primarily, the short message content and a caller ID). More importantly, attackers can spoof both the exact wording of the message and the caller ID of the real ones (see Section 5.2), making them less reliable cues. As such, VeriSMS explicitly avoids using any SMS cues but relies on inserted information (e.g., Message ID and Secret) to verify the message. At a higher level, our design is also inspired by prior studies on password managers [44, 58, 95]. Researchers find that users often do not use a password manager (correctly) to create unique and random passwords but instead use it to manage their weak/reused passwords [58]. This implies that a security mechanism, if not well-aligned with user habits, may not achieve the desired security impact. In our case, we envision that not all users will call to verify every message (validated by our user study), and thus we tailor the design to provide the basic level of security for users who do not call every time (Section 4.3).

Adaptive attacks. In Section 4.3, we have discussed adaptive attacks such as "calling the agent to extract user code," "code guessing attacks," and "omission attacks." We acknowledge that this is not an exhaustive list. More complicated adaptive attacks may directly manipulate the root of the trust. For example, (1) the attacker may send a message to the user to notify them that "the hospital has changed the voice agent's phone number" or (2) the attacker may send a message to notify the user that their secret words are reset to a different pair. However, we argue that the system can remain secure if users use the system correctly. Note that for both (1) and (2), the correct reaction is to use *VeriSMS* to verify these notification messages first before trusting the information. For example, a user can call the agent to verify the notification message about the "Secret Word Reset", which will reveal the notification's Message ID is invalid. It is also crucial for the organization that deployed VeriSMS to maintain a prolonged system with a consistent phone number, and educate users to always call the printed number on the card (instead of any other numbers) upon suspicion. That being said, we do recognize that educating users [35, 47] to correctly handle adaptive attacks is a non-trivial task and we leave further experimentation to future work.

Co-existing with Other Approaches. We believe VeriSMS is useful as an inclusive scheme to complement existing outreach channels. In other words, not everyone has to use VeriSMS but it is available if they do. For patients who already own and are familiar with smartphones, the dedicated patient portal apps (e.g., MyChart) could offer a secure outreach channel, as long as the apps are downloaded from trusted sources (e.g., the official app store). However, recent surveys show that many people such as older adults (i.e., age 65+) still use non-smartphones [2, 57], and only 61% of older adults own a smartphone [16]. Even for those that have a smartphone, certain older adults do not use a patient portal app [12]. VeriSMS provides an option for patients, especially those who only have access to (or use) basic phone calls and SMS. It can work jointly with other patients' communication channels, especially when other channels (e.g., patient portal) have more complex setup processes (e.g., account creation and login).

Generalization and Scalability. While we design VeriSMS for healthcare systems, it is worth discussing whether it can scale well if *multiple organizations and services* adopt it. From a user's perspective, if multiple services use VeriSMS (e.g., hospitals, banks, insurance companies), the user will need to keep track of the different secret words from different services (i.e., scalability issues). We believe the situation is more manageable (compared with conventional passwords). First, we envision only (a small number of) critical services (e.g. healthcare and financial services, insurance) needing a secure SMS-based outreach channel. It is not designed for (or needed by) general services. Second, SMS messages are often grouped under the service's phone number to form a message thread. Users are not required to perfectly memorize all the secret words—they can rely on the message threads to remind themselves about the secret words, check their consistency in the message thread, and call the voice agent to verify any inconsistent ones.

Limitations of the Study. We want to discuss and acknowledge a few limitations in our study. First, our user study is based on a small sample size. Given that our study requires participants to use *their personal mobile phones* over two time-gaped sessions, it is difficult to perform the study in an automated way at a large scale. As such, we focus our analysis on qualitative results to reveal the problems that exist and report descriptive statistics (e.g., for detection performance), rather than focusing on obtaining statistical power for quantitative analysis. Second, we explicitly informed participants during the consent phase that the system was designed to combat phishing. As a result, it's possible that participants are more prepared (than otherwise in real life) to detect potential phishing messages. This "priming" effect may have led to a better detection performance of users. Despite this limitation, we believe this is an acceptable approach used by many existing phishing studies [10, 26, 33, 43, 86, 89]. In our case, applying deception (e.g., sending phishing messages when users are unprepared) is especially challenging given that we need to test VeriSMS on the user's *personal* phone and we want to get their explicit consent upfront. Another source of the potential priming effect is the *tutorial stage* before the attack simulation (which may have led to better user performance). In practice, the hospital will introduce the system to patients with a similar instruction/tutorial (ecologically valid). In addition, our phase-2 result helps to mitigate the concerns about the priming effect of tutorials. Third, there are other threats to validity. For example, the link in the message is not clickable (to protect users), which may make users less worried about the potential harm of phishing. Another limitation is that we recruited participants from Prolific (from the U.S.), which may not be fully representative of the target patient population of the healthcare providers. For instance, participants recruited through Prolific all have basic computer skills (that certain older patients may not have in practice). As part of the future work, we plan to work with healthcare provider OSF Healthcare to perform internal tests with the target patient population.

9 CONCLUSION

In this paper, we design VeriSMS as an inclusive message verification method to address phishing and spoofing threats during SMS-based patient outreach. Through a user study, we confirm that users can correctly understand the system and use it to identify spoofed/phishing messages. The study also shows that VeriSMS has good-to-excellent usability and can significantly increase adversaries' costs. The result illustrates the importance of balancing security and usability to yield a practical solution that accommodates user habits.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions. This work was supported in part by NSF Award 2030521 and Jump ARCHES Award P336.

REFERENCES

[1] Mohamed Alsharnouby, Furkan Alaca, and Sonia Chiasson. 2015. Why phishing still works: User strategies for combating phishing attacks. *International Journal of Human-Computer Studies* 82 (2015), 69–82.

[2] Monica Anderson and Andrew Perrin. 2017. Technology use among seniors. https://www.pewresearch.org/internet/2017/05/17/technology-use-among-seniors/.

[3] Anon. 2023. Supplementary Materials for the Anonymous Submission. https://app.box.com/s/dtibikazdsjfifboocm8unruzo89kcme.

[4] Shahryar Baki and Rakesh M. Verma. 2023. Sixteen Years of Phishing User Studies: What Have We Learned? *IEEE Trans. Dependable Secur. Comput. (TDSC)* 20, 2 (2023), 1200–1212. https://doi.org/10.1109/TDSC.2022.3151103

[5] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101.

[6] John Brooke. 1995. SUS: A quick and dirty usability scale. *Usability Eval. Ind.* 189 (11 1995).

[7] Jianjun Chen, Vern Paxson, and Jian Jiang. 2020. Composition Kills: A Case Study of Email Sender Authentication. In *Proc. of USENIX Security.*

[8] Yixing Chen, Ju-Yeon Lee, Shrihari (Hari) Sridhar, Vikas Mittal, Katharine McCallister, and Amit G. Singal. 2020. Improving Cancer Outreach Effectiveness Through Targeting and Economic Assessments: Insights from a Randomized Field Experiment. *Journal of Marketing* 84, 3 (2020), 1–27. https://doi.org/10.1177/0022242920913025 arXiv:https://doi.org/10.1177/0022242920913025

[9] Soumyadeb Chowdhury, Ron Poet, and Lewis Mackenzie. 2014. A study of mnemonic image passwords. In *2014 Twelfth Annual International Conference on Privacy, Security and Trust.* 207–214. https://doi.org/10.1109/PST.2014.6890941

[10] Max Clasen, Fudong Li, and David Williams. 2021. Friend or Foe: An Investigation into Recipient Identification of SMS-Based Phishing. In *Human Aspects of Information Security and Assurance*, Steven Furnell and Nathan Clarke (Eds.). 148–163.

[11] Qian Cui, Guy-Vincent Jourdan, Gregor V. Bochmann, Russell Couturier, and Iosif-Viorel Onut. 2017. Tracking Phishing Attacks Over Time. In *Proc. of WWW.*

[12] Anthony D, Singer D, Kirch M, Solway E, Roberts S, Smith E, Hutchens L, Malani P, and Kullgren J. 2023. Use and Experiences with Patient Portals Among Older Adults. *University of Michigan National Poll on Healthy Aging* (2023). https://www.healthyagingpoll.org/reports-more/report/use-and-experiences-patient-portals-among-older-adults.

[13] Julie S. Downs, Mandy Holbrook, and Lorrie Faith Cranor. 2007. Behavioral Response to Phishing Risk. In *Proc. of eCrime.*

[14] Changlai Du, Hexuan Yu, Yang Xiao, Y. Thomas Hou, Angelos D. Keromytis, and Wenjing Lou. 2023. UCBlocker: Unwanted Call Blocking Using Anonymous Authentication. In *Proc. of USENIX Security.*

[15] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. 2008. You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *Proc. of CHI.*

[16] Michelle Faverio. 2022. Share of those 65 and older who are tech users has grown in the past decade. https://www.pewresearch.org/short-reads/2022/01/13/share-of-those-65-and-older-who-are-tech-users-has-grown-in-the-past-decade/.

[17] FCC. 2023. Combating Spoofed Robocalls with Caller ID Authentication. https://www.fcc.gov/call-authentication.

[18] Federal Trade Commission of the United States government 2022. Consumer Sentinel Network Data Book — ftc.gov. https://www.ftc.gov/system/files/ftc_gov/pdf/CSN-Data-Book-2022.pdf.

[19] FEITIAN Technologies US 2023. Time Based OTP Tokens (MS Azure). https://shop.ftsafe.us/collections/otp-time-based.

[20] Ian Fette, Norman Sadeh, and Anthony Tomasic. 2007. Learning to Detect Phishing Emails. In *Proc. of WWW.*

[21] Bree Fowler. 2023. Scam Texts Cost Consumers $330 Million in 2022, FTC Says. https://www.cnet.com/tech/services-and-software/scam-texts-cost-consumers-330-million-in-2022-ftc-says/.

[22] Samuel Gibbs. 2016. SS7 hack explained: What can you do about it? https://www.theguardian.com/technology/2016/apr/19/ss7-hack-explained-mobile-phone-vulnerability-snooping-texts-calls.

[23] Xiao Han, Nizar Kheir, and Davide Balzarotti. 2016. PhishEye: Live Monitoring of Sandboxed Phishing Kits. In *Proc. of CCS.*

[24] Grant Ho, Aashish Sharma, Mobin Javed, Vern Paxson, and David Wagner. 2017. Detecting Credential Spearphishing in Enterprise Settings. In *Proc. of USENIX Security.*

[25] Jason Hong. 2012. The State of Phishing Attacks. *Commun. ACM* 55, 1 (2012).

[26] Hang Hu and Gang Wang. 2018. End-to-End Measurements of Email Spoofing Attacks. In *Proc. of USENIX Security.*

[27] Sarah J. Iribarren, Kenrick Cato, Louise Falzon, and Patricia W. Stone. 2017. What is the economic evidence for mHealth? A systematic review of economic evaluations of mHealth solutions. *PLOS ONE* 12 (02 2017), e0170581. https://doi.org/10.1371/journal.pone.0170581

[28] IRS. 2022. IRS reports significant increase in texting scams; warns taxpayers to remain vigilant. https://www.irs.gov/newsroom/irs-reports-significant-increase-in-texting-scams-warns-taxpayers-to-remain-vigilant.

[29] Cristian Iuga, Jason RC Nurse, and Arnau Erola. 2016. Baiting the hook: factors impacting susceptibility to phishing attacks. *Human-centric Computing and Information Sciences* 6 (2016), 1–20.

[30] Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage. 2008. Spamalytics: An Empirical Analysis of Spam Marketing Conversion. In *Proc. of CCS.*

[31] Brian Krebs. 2018. SMS Phishing + Cardless ATM = Profit. https://krebsonsecurity.com/2018/11/sms-phishing-cardless-atm-profit.

[32] Ponnurangam Kumaraguru, Yong Rhee, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, and Elizabeth Nunge. 2007. Protecting People from Phishing: The Design and Evaluation of an Embedded Training Email System. In *Proc. of CHI.*

[33] Ponnurangam Kumaraguru, Yong Rhee, Steve Sheng, Sharique Hasan, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. 2007. Getting Users to Pay Attention to Anti-Phishing Education: Evaluation of Retention and Transfer. In *Proc. of eCrime.*

[34] Ponnurangam Kumaraguru, Steve Sheng, Alessandro Acquisti, Lorrie Faith Cranor, and Jason Hong. 2010. Teaching Johnny not to fall for phish. *ACM Transactions on Internet Technology* 10 (2010), 1–31.

[35] Elmer Lastdrager, Inés Carvajal Gallardo, Pieter Hartel, and Marianne Junger. 2017. How Effective is {Anti-Phishing} Training for Children?. In *Proc. of SOUPS.*

[36] Victor Le Pochat, Tom Van Goethem, and Wouter Joosen. 2019. Funny Accents: Exploring Genuine Interest in Internationalized Domain Names. In *Proc. of PAM.*

[37] Joel Lee, Lujo Bauer, and Michelle L. Mazurek. 2015. The Effectiveness of Security Images in Internet Banking. *IEEE Internet Computing* 19, 1 (2015), 54–62. https://doi.org/10.1109/MIC.2014.108

[38] Huichen Li, Xiaojun Xu, Chang Liu, Teng Ren, Kun Wu, Xuezhi Cao, Weinan Zhang, Yong Yu, and Dawn Song. 2018. A Machine Learning Approach to Prevent Malicious Calls over Telephony Networks. In *Proc. of IEEE S&P.*

[39] Eric Lin, Saul Greenberg, Eileah Trotter, David Ma, and John Aycock. 2011. Does Domain Highlighting Help People Identify Phishing Sites?. In *Proc. of CHI.*

[40] Baojun Liu, Chaoyi Lu, Zhou Li, Ying Liu, Hai-Xin Duan, Shuang Hao, and Zaifeng Zhang. 2018. A Reexamination of Internationalized Domain Names: The Good, the Bad and the Ugly. In *Proc. of DSN.*

[41] Mingxuan Liu, Yiming Zhang, Baojun Liu, Zhou Li, Haixin Duan, and Donghong Sun. 2021. Detecting and Characterizing SMS Spearphishing Attacks. In *Proc. of ACSAC.*

[42] Milena Soriano Marcolino, João Antonio Queiroz Oliveira, Marcelo D'Agostino, Antonio Luiz Ribeiro, Maria Beatriz Moreira Alkmim, and David Novillo-Ortiz. 2018. The Impact of mHealth Interventions: Systematic Review of Systematic Reviews. *JMIR Mhealth Uhealth* 6, 1 (17 Jan 2018), e23. https://doi.org/10.2196/mhealth.8873

[43] Claudio Marforio, Ramya Jayaram Masti, Claudio Soriente, Kari Kostiainen, and Srdjan Čapkun. 2016. Evaluation of Personalized Security Indicators as an Anti-Phishing Mechanism for Smartphone Applications. In *Proc. of CHI.*

[44] Peter Mayer, Collins W. Munyendo, Michelle L. Mazurek, and Adam J. Aviv. 2022. Why Users (Don't) Use Password Managers at a Large Educational Institution. In *Proc. of USENIX Security.*

[45] D. Kevin McGrath and Minaxi Gupta. 2008. Behind Phishing: An Examination of Phisher Modi Operandi. In *Proc. of LEET.*

[46] Sandhya Mishra and Devpriya Soni. 2021. DSmishSMS-A System to Detect Smishing SMS. *Neural Comput. Appl.* 35, 7 (jul 2021), 4975–4992. https://doi.org/10.1007/s00521-021-06305-y

[47] María M Moreno-Fernández, Fernando Blanco, Pablo Garaizar, and Helena Matute. 2017. Fishing for phishers. Improving Internet users' sensitivity to visual deception cues to prevent electronic fraud. *Computers in Human Behavior* 69 (2017), 421–436.

[48] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen. 1970. HOTP: An HMAC-based one-time password algorithm. https://www.rfc-editor.org/rfc/rfc4226.html.

[49] D. M'Raihi, S. Machani, M. Pei, and J. Rydell. 1970. TOTP: Time-based one-time password algorithm. https://www.rfc-editor.org/rfc/rfc6238.html.

[50] Hossen Mustafa, Wenyuan Xu, Ahmad-Reza Sadeghi, and Steffen Schulz. 2018. End-to-End Detection of Caller ID Spoofing Attacks. *IEEE Transactions on Dependable and Secure Computing* 15, 3 (2018), 423–436. https://doi.org/10.1109/TDSC.2016.2580509

[51] Yomna N. 2019. Gotta Catch 'Em All: Understanding How IMSI-Catchers Exploit Cell Networks. https://www.eff.org/wp/gotta-catch-em-all-understanding-how-imsi-catchers-exploit-cell-networks.

[52] Adam Oest, Yenganeh Safaei, Penghui Zhang, Brad Wardman, Kevin Tyers, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. 2020. PhishTime: Continuous Longitudinal Measurement of the Effectiveness of Anti-phishing Blacklists. In *Proc. of USENIX Security.*

[53] Adam Oest, Penghui Zhang, Brad Wardman, Eric Nunes, Jakub Burgis, Ali Zand, Kurt Thomas, Adam Doupé, and Gail-Joon Ahn. 2020. Sunrise to Sunset: Analyzing the End-to-end Life Cycle and Effectiveness of Phishing Attacks at Scale. In *Proc. of USENIX Security.*

[54] Daniela Oliveira, Harold Rocha, Huizi Yang, Donovan Ellis, Sandeep Dommaraju, Melis Muradoglu, Devon Weir, Adam Soliman, Tian Lin, and Natalie Ebner. 2017. Dissecting Spear Phishing Emails for Older vs Young Adults: On the Interplay of

Weapons of Influence and Life Domains in Predicting Susceptibility to Phishing. In *Proc. of CHI*.

[55] A. Orunsolu, O. Afolabi, S. Sodiya, and A. Akinwale. 2018. A Users' Awareness Study and Influence of Socio-Demography Perception of Anti-Phishing Security Tips. *Acta Informatica Pragensia* 7 (2018), 138–151.

[56] Sharbani Pandit, Krishanu Sarker, Roberto Perdisci, Mustaque Ahamad, and Diyi Yang. 2023. Combating Robocalls with Phone Virtual Assistant Mediated Interaction. In *Proc. of USENIX Security*.

[57] Carolyn Pang, Zhiqin Collin Wang, Joanna McGrenere, Rock Leung, Jiamin Dai, and Karyn Moffatt. 2021. Technology Adoption and Learning Preferences for Older Adults: Evolving Perceptions, Ongoing Challenges, and Emerging Design Opportunities. In *Proc. of CHI*.

[58] Sarah Pearman, Shikun Aerin Zhang, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2019. Why people (don't) use password managers effectively. In *Proc. of SOUPS*.

[59] Peng Peng, Chao Xu, Luke Quinn, Hang Hu, Bimal Viswanath, and Gang Wang. 2019. What Happens After You Leak Your Password: Understanding Credential Sharing on Phishing Sites. In *Proc. of Asia CCS*.

[60] Peng Peng, Limin Yang, Linhai Song, and Gang Wang. 2019. Opening the Blackbox of VirusTotal: Analyzing Online Phishing Scan Engines. In *Proc. of IMC*.

[61] Jon Peterson, Henning Schulzrinne, and Hannes Tschofenig. 2014. Secure Telephone Identity Problem Statement and Requirements. RFC 7340. https://doi.org/10.17487/RFC7340

[62] Jon Peterson and Sean Turner. 2018. Secure Telephone Identity Credentials: Certificates. RFC 8226. https://doi.org/10.17487/RFC8226

[63] Pawan Prakash, Manish Kumar, Ramana Rao Kompella, and Minaxi Gupta. 2010. Phishnet: Predictive Blacklisting to Detect Phishing Attacks. In *Proc. of INFOCOM*.

[64] Sathvik Prasad, Elijah Bouma-Sims, Athishay Kiran Mylappan, and Bradley Reaves. 2020. Who's Calling? Characterizing Robocalls through Audio and Metadata Analysis. In *Proc. of USENIX Security*.

[65] Sathvik Prasad, Trevor Dunlap, Alexander Ross, and Bradley Reaves. 2023. Diving into Robocall Content with SnorCall. In *Proc. of USENIX Security*.

[66] Protectimus Limited 2022. Two factor authentication hardware TOTP token. https://www.protectimus.com/protectimus-two.

[67] Peter Rebeiro, Giorgos Bakoyannis, Beverly Musick, Ronald Braithwaite, Kara Wools-Kaloustian, Winstone Nyandiko, Fatuma Some, Paula Braitstein, and Constantin Yiannoutsos. 2017. An Observational Study of the Effect of Patient Outreach on Return to Care: The Earlier the Better. *Journal of acquired immune deficiency syndromes (1999)* 76 (2017), 141–148. https://doi.org/10.1097/QAI.0000000000001474

[68] Joshua Reynolds, Deepak Kumar, Zane Ma, Rohan Subramanian, Meishan Wu, Martin Shelton, Joshua Mason, Emily Stark, and Michael Bailey. 2020. Measuring Identity Confusion with Uniform Resource Locators. In *Proc. of CHI*.

[69] RoboKiller. 2021. The Robocall Report 2021 MID-YEAR PHONE SCAM REPORT. https://www.robokiller.com/spam-text-insights/..

[70] Merve Sahin, Aurélien Francillon, Payas Gupta, and Mustaque Ahamad. 2017. SoK: Fraud in Telephony Networks. In *Proc. of EuroS&P*.

[71] Dawn M Sarno, Joanna E Lewis, Corey J Bohil, and Mark B Neider. 2020. Which phish is on the hook? Phishing vulnerability for older versus younger adults. *Human factors* 62, 5 (2020), 704–717.

[72] Stuart E. Schechter, Rachna Dhamija, Andy Ozment, and Ian Fischer. 2007. The Emperor's New Security Indicators. In *Proc. of IEEE S&P*.

[73] A Shaik, R Borgaonkar, N Asokan, V Niemi, and J Seifert. 2017. Practical attacks against privacy and availability in 4G/LTE mobile communication systems. In *Proc. of NDSS*.

[74] Sarah Stewart de Ramirez, Jeremy McGarvey, Abby Lotz, Mackenzie McGee, Tenille Oderwald, Katherine Floess, Roopa Foulger, Melinda Cooling, and Jonathan A. Handler. 2022. Closing the Gap: A Comparison of Engagement Interventions to Achieve Equitable Breast Cancer Screening in Rural Illinois. *Population Health Management* 25, 2 (April 2022), 244–253. https://doi.org/10.1089/pop.2021.0382

[75] Christopher Thompson, Martin Shelton, Emily Stark, Maximilian Walker, Emily Schechter, and Adrienne Porter Felt. 2019. The Web's Identity Crisis: Understanding the Effectiveness of Website Identity Indicators. In *Proc. of USENIX Security*.

[76] Christopher Thompson, Martin Shelton, Emily Stark, Maximilian Walker, Emily Schechter, and Adrienne Porter Felt. 2019. The web's identity crisis: understanding the effectiveness of website identity indicators. In *Proc. of USENIX Security*.

[77] Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn. 2019. Users Really Do Answer Telephone Scams. In *Proc. of USENIX Security*.

[78] Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn. 2016. SoK: Everyone Hates Robocalls: A Survey of Techniques Against Telephone Spam. In *Proc. of IEEE S&P*.

[79] Twilio. [n. d.]. Respond to Incoming Phone Calls in Python. https://www.twilio.com/docs/voice/tutorials/how-to-respond-to-incoming-phone-calls/python.

[80] usability.gov. 2013. System Usability Scale (SUS). https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[81] Amber van der Heijden and Luca Allodi. 2019. Cognitive Triaging of Phishing Attacks. In *Proc. of USENIX Security*.

[82] Javier Vargas, Alejandro Correa Bahnsen, Sergio Villegas, and Daniel Ingevaldson. 2016. Knowing your enemies: leveraging data analysis to expose phishing patterns against a major US financial institution. In *Proc. of eCrime*.

[83] Verizon. 2020. 2020 Mobile Security Index Report. Verizon. https://enterprise.verizon.com/resources/reports/2020-msi-report.pdf. Accessed: 10 October 2021.

[84] Verizon. 2022. Call Filter. Verizon. https://www.verizon.com/business/products/contact-center-cx-solutions/voice-security/call-filter/.

[85] Jingguo Wang, Yuan Li, and H Raghav Rao. 2016. Overconfidence in phishing email detection. *Journal of the Association for Information Systems* 17 (2016), 1.

[86] Zikai Alex Wen, Zhiqiu Lin, Rowena Chen, and Erik Andersen. 2019. What.Hack: Engaging Anti-Phishing Training Through a Role-Playing Phishing Simulation Game. In *Proc. of CHI*.

[87] Wikipedia. 2023. SiteKey. https://en.wikipedia.org/wiki/SiteKey.

[88] Ryan T. Wright, Matthew L. Jensen, Jason Bennett Thatcher, Michael Dinger, and Kent Marett. 2014. Research Note—Influence Techniques in Phishing Attacks: An Examination of Vulnerability and Resistance. *Information systems research* 25 (2014), 385–400.

[89] Aiping Xiong, Robert W Proctor, Weining Yang, and Ninghui Li. 2019. Embedding training within warnings improves skills of identifying phishing webpages. *Human Factors* 61, 4 (2019), 577–595. https://doi.org/10.1177/0018720818810942

[90] Guanhua Yan, Stephan Eidenbenz, and Emanuele Galli. 2009. SMS-Watchdog: Profiling Social Behaviors of SMS Users for Anomaly Detection. In *Proc. of RAID*, Engin Kirda, Somesh Jha, and Davide Balzarotti (Eds.).

[91] Jim Youll. 2006. Fraud Vulnerabilities in SiteKey Security at Bank of America. *CR-Labs* (2006). https://www.cr-labs.com/publications/SiteKey-20060718.pdf.

[92] Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong. 2007. Phinding Phish: Evaluating Anti-Phishing Tools. In *Proc. of NDSS*.

[93] Yue Zhang, Jason I Hong, and Lorrie F Cranor. 2007. Cantina: a content-based approach to detecting phishing web sites. In *Proc. of WWW*.

[94] Yiming Zhang, Baojun Liu, Chaoyi Lu, Zhou Li, Haixin Duan, Shuang Hao, Mingxuan Liu, Ying Liu, Dong Wang, and Qiang Li. 2020. Lies in the Air: Characterizing Fake-Base-Station Spam Ecosystem in China. In *Proc. of CCS*.

[95] Samira Zibaei, Dinah Rinoa Malapaya, Benjamin Mercier, Amirali Salehi-Abari, and Julie Thorpe. 2022. Do Password Managers Nudge Secure (Random) Passwords?. In *Proc. of SOUPS*.

A EXPLORATORY STUDY DETAILS

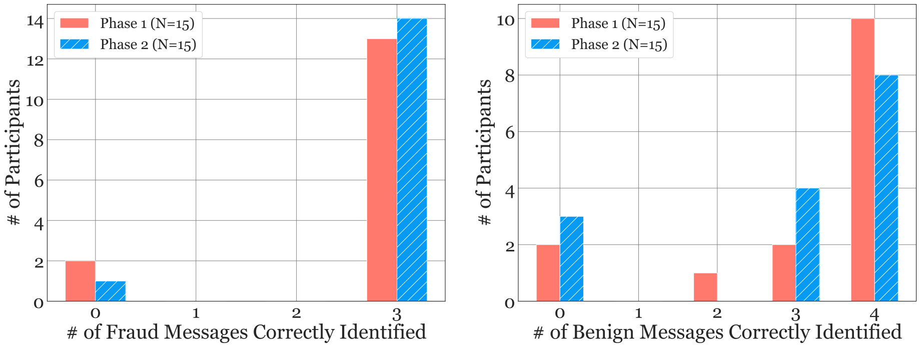
In this section, we report the data analysis results of the *exploratory study* with N=15 participants. All participants have attended both phase-1 and phase-2. The purpose of the analysis is to serve as a reference for the *validation study* presented in Section 7, to show that the problems revealed in the exploratory study have been addressed by the revision.

Classification Performance. The message classification performance on benign and fraudulent messages is shown in Fig. 6. The overall performance is worse compared to the validation study, which corresponds to the design problems we identified in Section 7. As shown in Fig. 6a, 2 participants (12%) failed to identify all the fraudulent messages during phase-1, and 1 participant (6%) remained unable to identify all the fraudulent messages during phase-2. As shown in Fig. 6b, there are also 5 participants (33%) who failed to correctly identify some or all benign messages during phase-1, and 6 participants (40%) failed during phase-2.

Usability Scores. We also analyzed the responses collected during the exit survey of the exploratory study for usability scores.

Fig. 7 reflects the SUS scores of N=15 participants' responses to the system usability scale (SUS) questionnaire during phase-1 and phase-2, respectively. The mean SUS score is 75.33 for phase-1, with a median SUS score of 77.5. The mean SUS score for phase-2 is 78.5, with a median SUS score of 82.5.

Overall, the fact that all these results are clearly improved in the revised system in the validation study (see Section 7) confirms the effectiveness of the revision.



(a) Detected Fraud Messages **(b) Detected Benign Messages**

Figure 6: Message Detection Results—Correctness of message classification by participants in the exploratory study.

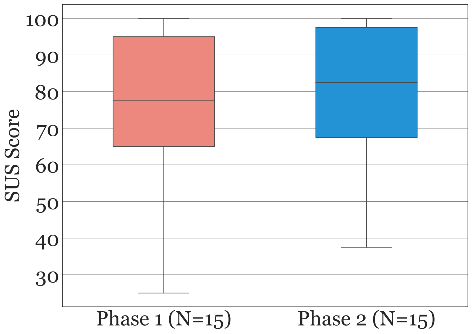


Figure 7: Usability Results—The SUS score of each participant in the exploratory study.

Study Stage	**Theme**	**Code**
Tutorial	Confusions	about choosing secret about Message ID about instruction about phone number
	Usability comments	about switching screens about insufficient time for input about voice prompts
Attack Sim.	Confusions	about choosing secret about Message ID about instruction about phone number
	Usability comments	about switching screens about insufficient time for input about voice prompts
Exit Interview	Self-reported confidence	score from 1 to 5
	Strategy for calling voice agent	never call (rely on secret word) never call (rely on language and/or content) call when content is suspicious call when content is important (context-related / before interaction) always call
	Strategy for changing secrets	never change change periodically change if has difficulty remembering change if known data breach happens change upon receiving a fraud message
	Other suggestions	direct quote

Table 3: Codebook—We list the high-level codes in the codebook used in our study.

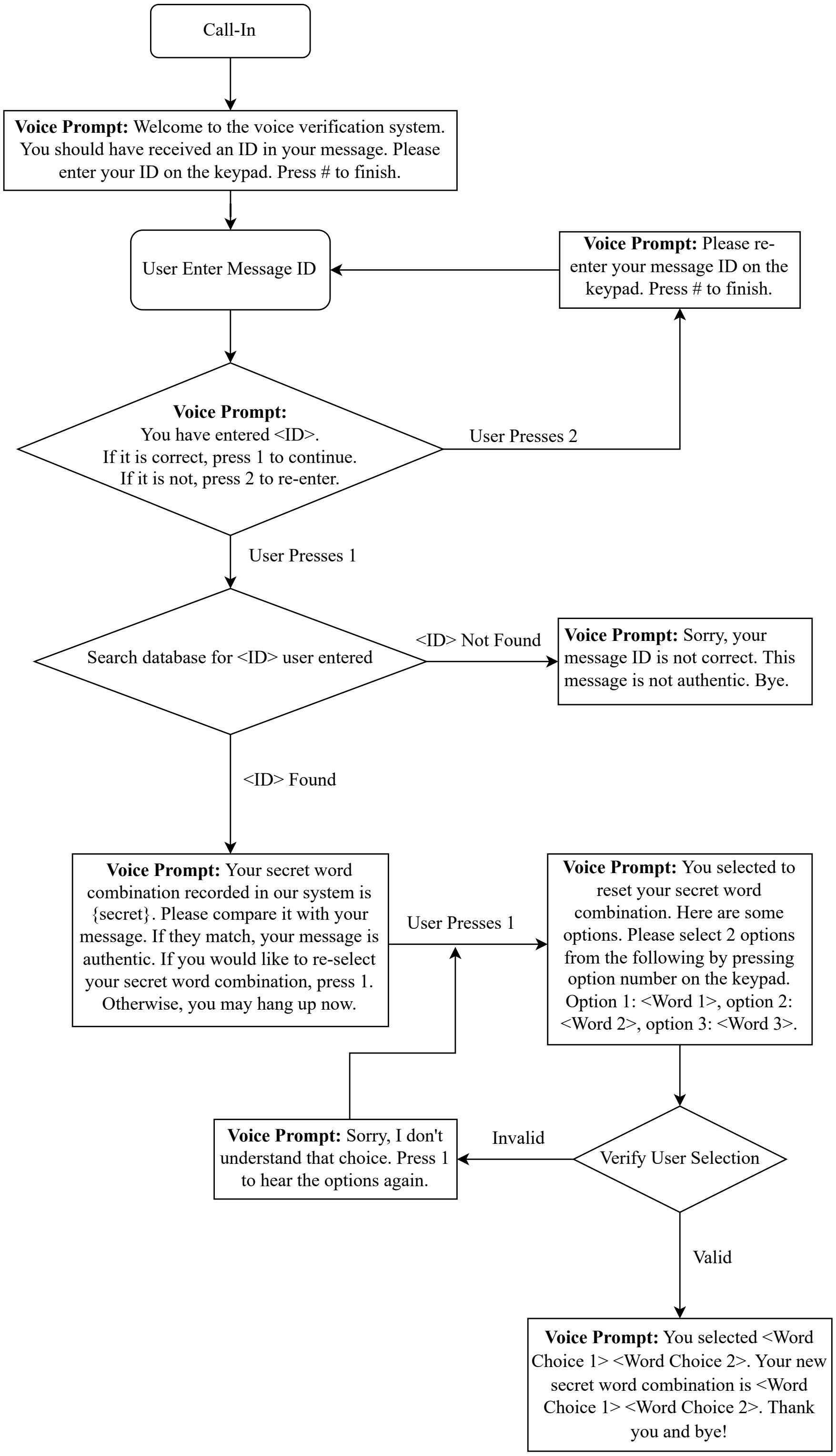

Call-In
Voice Prompt: Welcome to the voice verification system. You should have received an ID in your message. Please enter your ID on the keypad. Press # to finish.
User Enter Message ID
Voice Prompt: Please re-enter your message ID on the keypad. Press # to finish.
Voice Prompt: You have entered <ID>. If it is correct, press 1 to continue. If it is not, press 2 to re-enter.
User Presses 2
User Presses 1
Search database for <ID> user entered
<ID> Not Found
Voice Prompt: Sorry, your message ID is not correct. This message is not authentic. Bye.
<ID> Found
Voice Prompt: Your secret word combination recorded in our system is {secret}. Please compare it with your message. If they match, your message is authentic. If you would like to re-select your secret word combination, press 1. Otherwise, you may hang up now.
User Presses 1
Voice Prompt: You selected to reset your secret word combination. Here are some options. Please select 2 options from the following by pressing option number on the keypad. Option 1: <Word 1>, option 2: <Word 2>, option 3: <Word 3>.
Voice Prompt: Sorry, I don't understand that choice. Press 1 to hear the options again.
Invalid
Verify User Selection
Valid
Voice Prompt: You selected <Word Choice 1> <Word Choice 2>. Your new secret word combination is <Word Choice 1> <Word Choice 2>. Thank you and bye!

Figure 8: Workflow of the Voice Agent—The chart shows how a caller interacts with the voice agent in the revised version of VeriSMS.