End-to-End Call Authorization in Public Telephony

Sam (Joel Samper), Aspen Olmsted
New York University, USA

Abstract

More than a decade of papers talking about preventing SPIT (SPam over Internet Telephony), yet we still receive spam calls these days. We recover here an old proposal about issuing secret codes to potential callers, which they are to provide on each call in order to prove legitimacy. As an up-to-date contribution, we suggest an out of band mechanism to exchange and validate those secrets, based on mobile apps and web services, and we leverage TOTP (Time-based One Time Passwords) to improve our recipe. Our goal is to protect mobile subscribers from unwanted calls (not only spam), and at the same time to allow for anonymous calls. Is that feasible?

1. Introduction

In 2020, 58% of frauds reported to the US Federal Trade Commission used phone calls or texts as the contact method, with a loss of $526M [1]. Mobile phones are the preferred means for scam and spam calls compared to a landline, and this trend has been noticeable over the recent years. Also, the percentage of calls to cell phones that are fraudulent increased, roughly 4% in 2017 to 29% in 2018 [2] [3]. This increase is before the COVID-19 outbreak. Consumer trends on household lines show that mobile is replacing landlines [4].

We do answer spam calls [5]. Unlike emails or instant messages, synchronous phone calls are meant to be responded at the very moment they happen. Because we occasionally receive meaningful calls from people we do not know beforehand, we may feel the urge to answer even if the calling number is unrecognized. For example, our pizza delivery person arrived, or we just got awarded a Nobel Prize, or whatever is the case. This happens even if, annoyingly, we receive them during our personal activities.

Aside from robocalls, we should also consider what we do with unwanted calls from real people, as it may be the case of stalking. Even if the number is blocked or if anonymous calls are rejected, the offender may always get a new number, so we are back to square zero. We can report to the police, but if the offender is clever, they might remain unidentified. Otherwise, we are forced to block calls from unknown numbers (which impacts legitimate calls), or simply change our number (mainly burdensome).

It is known that, unlike common-use email anti-spam techniques, there are no simple ways to prevent SPIT. Aside from the technical limitations, TSPs (Telephony Service Providers) are not motivated to implement anti-robocall technologies [7]. A new system implemented on the carrier side would require a return on investment analysis, either by charging extras for the service, by getting more customers as a result of an innovative service, or as a means to avoid fines imposed by legislation.

Despite the success of VoIP, public telephony is not going to disappear overnight. Moreover, the current lack of reliability of packet-switched networks make them unsuitable for emergency calls. So, for now, internet and telephony coexist, leaving users with no option but to keep having a telephone number, and letting network engineers maintain and improve the PSTN (Public Switched Telephone Network) to the best of their abilities.

K. Ono and H. Schulzrinne [6] proposed using weak social ties interactions via web, email or social network services, to exchange Weakly Secret Information [7] that potential caller would later use to prove that their calls are legitimate. They also suggested using web interactions to whitelist automatically. Such proposal could be further classified as No Interactions With Call Participants [8] [9], and Non-Intrusive [10]. This paper will suggest an extension of their proposal, which we dubbed Authorization Secret, for mobile recipients equipped with a smartphone. We will define how an app, coupled with a web service, may act as a single point for the initial exchange of the secret; we will show how TOTP can be leveraged here; we will analyze how to integrate the proposal technically with non-IP telephony access protocols and SS7 international trunks of the PSTN; and we will self-assess its usability, deployability, effectiveness, security and design properties, with some final words acknowledging downsides and limitations.

2. Authorization Secret Proposal

We introduce here a proposal that enables call authorization in public telephony when the recipients are mobile users with smartphones, although we extended it to other use cases. This solution is based on a 13-digit decimal Authorization Secret, also referred to as ‘Secret’. Recipients issue Secrets to potential callers or groups of callers.

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

In addition to dialing the recipient phone number, callers must include that Secret in outgoing calls, when this feature is enabled. Each Secret is composed of a 5-digit ID, plus an 8-digit Code, as shown in "Figure 1". The Code may be either Static (it does not change once issued) or TOTP-based (it is generated at call time with a cryptographic algorithm, using a key and the current timestamp as inputs) [11].

For TOTP-based individually-assigned Codes, a counter is appended to the timestamp used as input for the algorithm, so that each call has a different non-reusable Code. For example, if the TOTP timer is 30 seconds, a 1-to-30 counter allows to place calls every second from the same sender to the same recipient (no perceived throttling). The counter is incremented in each call and is reset whenever the TOTP timer expires. This technique prevents reusing eavesdropped TOTP Codes.

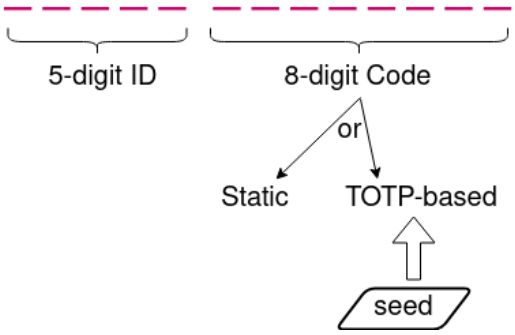

Figure 1. Structure of the 13-Digit Secret

A typical use case would be sharing your number with friends, family, or companies, for them to be able to call you. To make use of this feature, the recipient must be equipped with an upgraded smartphone, and the network components must support it. Otherwise, the call will have to be processed normally in Legacy mode, as we will explain below. This proposal is not an all-or-nothing model, but a flexible one. It is backward compatible because it is and optional feature offered by the telephony network, enabled on a per user basis and even on a per call basis.

2. Modes of Operation

The recipient device can be configured with one of these three modes of operation, as shown in Figure 2:

• Legacy: The recipient processes incoming calls without applying any call blocking based on the Secret, regardless of whether the caller sends it or not. Even on Legacy mode, the smartphone could show if the call had a valid Secret, as this may be useful during a transition phase or for troubleshooting purposes.

• Flexible: The incoming Secret will be processed when present. If a call has a present but invalid Secret, it will be rejected. If the Secret is not present, the call may be accepted under certain conditions, depending on the set of policies configured on the user device, for example:

◦ if the calling number is a number present in the phone agenda. Alternatively, the call could be sent to a second stage for a CAPTCHA/Turing test or other filtering techniques.

◦ if the calling number is a number that has been dialed recently.

• Strict: Only calls with a present and valid Secret are accepted.

Emergency calls are always accepted, no matter what.

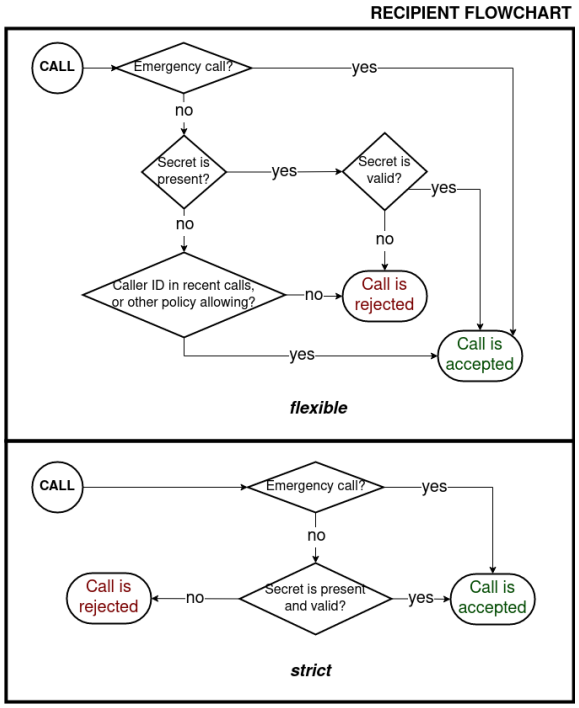

Figure 2. Flexible and Strict Modes of Operation on the Recipient Side

All modes of operation are independent of how the recipient's device makes calls itself. For example, a device may be configured in Legacy mode, so that it will not process any Secrets for incoming calls, but at the same time it may make Secret-enabled calls. And vice-versa. The use of the Authorization Secret feature on the caller side relies exclusively on the user who calls, provided that both

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

the calling device and the network support and enable it. The acceptance of calls depends exclusively on the recipient user. The network is only responsible of transporting the Secret from caller to recipient.

2.1. Issuing Secrets

An app on the recipient device is always the starting point for issuing Secrets. Both the ID and Code components are generated at random. The software will make sure to pick a new ID. The ID and Code pairs are stored in the recipient device's local DB (database) of issued Secrets, along with contextual information such as timestamp, location, a personal text note, etc. It is recommended, though not mandatory, to link the Secret with a contact entry in the phone's agenda of the recipient's contacts. Likewise, the calling party must have a DB of received Secrets, each of which will have an associated phone number, usually linked with a contact in the phone agenda. Therefore, each device will have a local DB of issued Secrets, plus a local DB of received Secrets.

We define four methods for passing on the phone number to someone else along with the corresponding Secret, as shown in Figure 3.

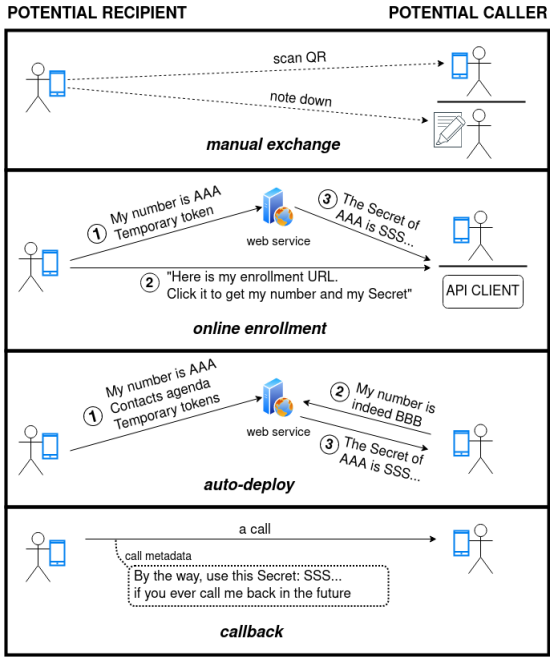

Figure 3. Methods for issuing the Secret to Potential Callers

Manual Exchange. The Secret is displayed on the recipient device, who shows it to a potential caller for them to note it manually. For TOTP-based Codes, it is necessary to pass on the seed passphrase, or to display it with a QR. For manual exchange of Static Codes in a standard way, we could embed the Secret within the recipient phone number by prepending two star signs (**) and appending the 13-digit Secret. For example, let's say the subscriber number is +10123456789. Then, the number passed onto the potential caller would be **+10123456789SSS, where SSS is the Secret number. This subscriber number in long format is what the caller would keep in their agenda, and also the number to be dialed, provided that the line supports it. Note that the choice of sending ** at the beginning of the dialing sequence avoids delay issues with two-stage dialing. Otherwise, when dialing +10123456789, the user would experience a delay in the call setup, because the network would not able to determine a priori if this is a completed number without Secret, or an incomplete number with Secret.

Online Enrollment. The app first generates an ID and a temporary random Token sent to an online service linked to a user account. An enrollment link to that service is generated, which includes the phone number, the ID, and the Token, all of them embedded in the URL. This link is passed to the potential caller using email, instant messaging, or any other means. Once they click the link, they can choose to download a Static or a TOTP-based Code, all generated and synchronized by the online service, or previously set by the recipient app. For online enrollment, a standardized web service needs to be defined to ensure interoperability across applications.

By default, the Token should be automatically deactivated after a reasonable timeout or once the potential caller has completed the enrollment. Otherwise, it can be shared with a group of people. In some scenarios, the user will want to share a single enrollment URL with multiple parties but assign a different Secret to each of them. Ideally, the recipient would have to confirm each enrollment request from the app or impose some limits to avoid abuse. Assigning individualized Secrets allows granularity for revoking. At the same time, the user would also be able to revoke all Secrets that originated from that enrollment URL, if desired, as that would be kept as metadata of the issued Secret. In other scenarios, the user may wish that multiple people use exactly the same Secret.

Auto-deploy. The recipient user enables the app to automatically assign an individual temporary Token to each phone contact present in their agenda. When callers use the web service, it will ensure that they own the phone number they claim to have by sending an SMS with a random password and making them confirm it. The online service could also use algorithms that check how many contacts they have in common in their respective agendas, to double-check social ties. The approach is similar to the social network service use case explained in [6], but we never rely on the caller ID, and an individual Secret is issued to each potential caller.

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

Callback. The protocol could allow the caller party to send a Secret of their own. Then, right before placing the call or during the call, the caller could press an option to generate and send a Secret. The network protocol should therefore allow two fields: one for the Secret of the calling party and one for the Secret of the called party. They would both be optional and mutually-independent fields. A possible default is to automatically send the own Secret for calls to numbers stored in the agenda. This method is somewhat similar to HTTP Message Exchanges in [6].

2.2. Placing calls

The calling party must send a valid Secret, in addition to dialing the phone number, when making a call to a recipient who requires Authorization. The calling party has three options, depending on the device and type of line, as shown in Figure 4.

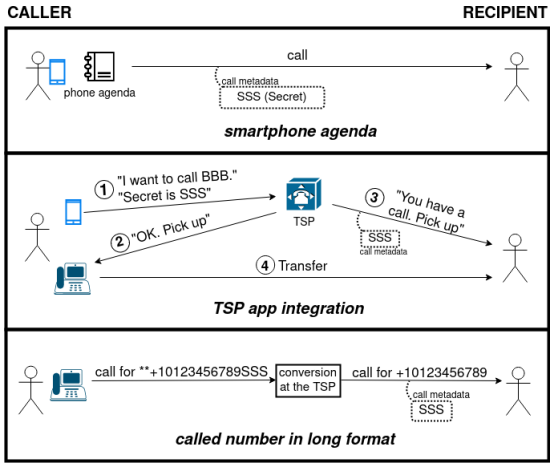

Figure 4. Methods for placing Secret-enabled Calls

• If calling from a smartphone or some intelligent system, the calling party can dial from the agenda of contacts. If the contact entry has an associated Secret information, everything is handled transparently. Otherwise, the dial interface should have a separate field to introduce the Secret manually.

• If calling from a landline, the user could use an app coupled with the TSP to trigger the call out of the band (via internet). The app would send the desired called number and its Secret to the TSP at call time. The TSP would then ring the caller's phone, and transfer that call to the called number along with the Secret.

• The user could store the number in long format in the phone memory for any system, if the Code is Static. This approach is feasible if the phone has a memory feature that allows long numbers and the star sign. Otherwise, the user would have just to manually dial the Secret.

2.3. Receiving and Checking

Depending on the mode of operation configured on the recipient device, the received Secret will be processed or not. If so, it will extract the ID portion of the Secret, and check matching entries in its local DB of issued Secrets. If a match is found, it will verify if the Code is valid. If the Code is valid, the call is accepted and processed normally. Otherwise, the call may be generically refused, or explicitly rejected indicating an Authorization Failed error code. Note that other features, such as calling number blacklisting, may eventually refuse the call even if the Code is correct.

2.4. Revoking

Revoking a previously granted Secret should be a simple procedure for a recipient; for example, if linked with a contact, they can look up the contact in the agenda and revoke the authorization. However, this does not necessarily imply deleting the contact number or losing the ability to call that contact. If not linked with a contact, the user should access the local DB of issued Secrets and use other metadata (date and time, personal notes, etc.) to look up the entry. The action of revocation is not notified to the network, nor to the contact. Nevertheless, it has immediate effect.

Secrets can be revoked at any time, therefore unwanted calls are easy to prevent from happening again. This prevention is true as long as the user has issued a different Secret to each contact or use case. The smartphone interface should deter users from reusing Secrets with multiple contacts.

Our proposal allows to share your number in forums, web sites or chats in a safer way. A use case scenario is a small business user who runs several businesses and publishes their personal number with a different Secret for each business. If one of the ventures shuts down, the user then revokes the associated Secret. In this case, the authorization grant is linked to the reason of the call. You can replace business with 'project', 'special hobby', or anything you like. Without the Secret feature, once you give a number you have no control over how it will be used.

2.5. Other mobile services

We will address here mobile services other than regular calls. For SMS (Short Message Service), we can apply a similar approach than we have done with the calls. We could assign a specific identifier for an Information Element within the TP-UD field to hold the Secret, following Specification 23.040 of the 3GPP [12]. Because the TP-Service-Centre-Time-

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

Stamp mandatory field holds the time when the network received the message, this information must be used in the TOTP algorithm.

In the case of voicemail, it is equally necessary to adapt such scenario for Authorization Secret compliance. Otherwise, the recipient will have to listen to any unwanted messages left in the inbox. The voicemail service of the TSP should collect the received Secret as metadata of the message. It is advisable to add a default warning to the caller, in addition to the user-defined greeting, informing that the message will be silently discarded if it does not pass Authorization. Once the recipient phone is online, the TSP will include the Secret in the Voicemail Message Waiting the same way as in SMS. If the phone sends an Authorization Failed error code defined in the Specification, the TSP must delete the voice message.

In call forwarding, maybe a drastic solution would be to just strip the Secret when forwarding the call, expecting the final recipient to accept forwarded calls by default. Otherwise, a more advanced mechanism for Authorization Secret transitivity should be designed, such as sharing a read-only copy of the Secrets DB.

If roaming to an area where the Secret feature is unsupported, the phone would be aware of that by interacting with the mobile operator. In such a scenario, it would temporarily switch to Legacy mode and signal so to the user with a pop-up message or a screen icon.

3. Technical Requirements

In this section we will discuss some technical requirements and implementation aspects of our proposal.

3.1. Inter-carrier trunks

How to integrate the call Authorization Secret across PSTN inter-carrier or international trunks? One possibility could be to embed it with the called party number, using the long format explained above. However, this is technically not possible with the current protocol standards. As part of the SS7 set of standards, ITU Q.767 Recommendation [13] defines the IAM (Initial Address Message) as the primary message type used to initiate calls. The called party number parameter has a maximum length of 11 octets, which allows for 18 digits. This length does not leave enough room for the Authorization Secret. Moreover, the E.164 Recommendation [14] is used for the numbering plan, with a maximum of 15 digits excluding the international prefix. Only decimal digits are allowed.

We suggest using a User-to-User Information optional parameter (which has a maximum of 128 octets) in the IAM message. Our 13-digit Authorization Secret could be placed there, using a similar digit encoding as the called party number defined in section C.3.7 of ITU Q.767. All TSPs should perform appropriate conversions between long-format and short-format numbers, extracting and inserting the Secret as a User-to-User Information parameter across their trunks to other providers. For the callback method, a similar field should be used to hold the callback information.

In addition to that, we would have to add an Authorization Failed cause code to the ITU Q.850 Recommendation [15].

3.2. Calling party's subscriber lines

In order to ensure system interoperability, all access protocols offered by TSPs should aim to at least enable sending the Secret for all subscriber-to-network calls, either in long format numbering or using special fields. They should also perform the necessary conversions before forwarding the call to other providers. If some provider already uses ** for other special services, then it may accommodate a different service code (such as ##) for their customers, to avoid overlapping. Here is an abridged technical overview of how some widespread access protocols used in telephony could accommodate a unique field for the Secret, or the star sign for long format dialing. See Table 1 for a summary.

• POTS lines support the star key with tone dialing, by using the 941 Hz + 1209 Hz multifrequency tone [16].

• SIP supports the star sign as a valid character in the user part of the URI in the INVITE message, as defined in RFC 3261 [17]. Note that the hash sign (#) is not supported, hence choosing of the star (*) for better consistency across protocols. This method makes it theoretically feasible to dial the Secret from end-user SIP phones without updating the SIP stack on those. Alternatively, the Secret could be sent in a separate header, as suggested in [6].

• ISDN supports the star sign in the called party field of the SETUP message with its Q.931 specification [18], using the 2/10 character in coded representation defined in ITU's T.50 Recommendation (formerly known as IA5). When dialing using the Keypad facility, the maximum number length is 34 characters. Alternatively, the protocol could be slightly adapted to hold the Secret in a specially-purposed Information Element.

• H.323/H.225 also follows the Q.931 specification, so the above paragraph applies.

• For mobile phone subscribers of generations 2G through 5G, we can use the Calling Party Subaddress

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

and Called Party Subaddress fields (40 digits maximum, each) to hold the Secret values. Alternatively, can define a new User-User Information Element (35 octets maximum) in the SETUP message as defined in Specification 24.008 of the 3GPP [19]. This approach applies both if the call originates or terminates at the user's Mobile Station.

It should be noted that with ISDN's Keypad facility limitations, we have 34 – 2 [for the **] – 13 [Secret length] – 15 [E.164 limit] = 4 digits as space left for the international prefix. We have only found instances of larger international prefixes in Colombia, Finland, Indonesia, South Korea, and Thailand. Also, when dialing a call within the same country, the resulting number must be no longer than 34 – 2 [for the **] – 13 [Secret length] = 19 digits, but we have not been able to find any problematic case.

In all subscriber lines protocols, a new Authorization Failed cause code should be defined for recipients to signal when the received Secret is not correct. This code should map to its equivalent in SS7.

Table 1. Implementation Options on Telephony Protocols

	Base protocol	***Protocol message***	***Field or Feature***
POTS	analog	-	TSP app integration (out of band)
POTS	analog	-	Star key (MF tone)
SIP	RFC 3261	INVITE	URI user part, or a new header
ISDN / H.323	Q.931	SETUP	Called Party field, or Information Element
2G-5G	3GPP	SETUP	Calling/Called Party Subaddress, or User-User Information Element
Inter-carrier trunks	SS7	IAM	User-User Information

3.3. Feature support in mobile telephony

Every time a mobile device registers to a network, it should receive a notification indicating that the carrier supports the Authorization Secret (if that is the case). If the network does not support it but the device does, then the recipient device will use legacy protocols in all outgoing calls. For incoming calls, it would temporarily switch to Legacy mode. Also, when a Secret-compatible recipient device registers to a network that has signaled support, the device will have to signal appropriately to the network. The signal is also required when the subscriber's device changes the Authorization Secret mode of operation while registered to a network with support. When delivering a call, the network will send the Secret only to compatible devices.

3.4. Smartphones

Smartphones should be smart. Aside from the necessary adaptations to interoperate with the new mobile telephony protocols, they will need other upgrades. First, phones must have a local DB of issued and received Secrets, with a user interface to manage it, including the generation of new Secrets. Second, the phone will have to apply the configured authorization policies and check the Secret on all incoming calls if enabled. Also, when dialing manually, the phone should have a separate field for entering the Secret. And third, a system service or third-party app is deployed to exchange Secrets conveniently.

3.5. Phone number fields

It is common for users to introduce the phone number in web forms, applications or personal/corporate directories. Some websites and databases enforce the phone number to have only decimal digits or do some sort of input validation. In such cases, the corresponding fields in the database, web forms or any other interface will have to be updated to accommodate the star sign and longer-than-usual numbers. This would allow recipients to provide a Static Secret. However, instead of that, we recommend provisioning a unique field for the enrollment URL and process it appropriately to get the phone number and the Secret. For convenience, browsers should have a plugin that integrates everything in a single click.

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

4. Self-Assessment

We will do a self-assessment of this proposal's usability, deployability, and robustness properties to the best of our abilities, mostly using the SoK in [7]. We will also discuss some design and security properties. Finally, we will identify downsides and limitations.

4.1. Usability

Once the the network and the devices are Authorization Secret compliant, and the Secrets have been exchanged, we have the properties of No-Disturbance-to-Recipient, Scalable-for-Recipient, and Permissive-for-VoIP-Callers. Before that, there will be a period of transition where recipients, legitimate callers and telephony infrastructure providers, will have to upgrade systems and change configurations. Our proposal is deemed to have good usability by initially leveraging the auto-deploy mode. The user may have to manually issue Secrets with contacts who do not have internet access, though.

In calls where the number is dialed manually, the user will have to dial 15 more digits, taking the Legacy mode as the base case. Therefore, it is not completely Effortless-for-Caller or Negligible-Delays. However, since many calls are made from contact agendas or from databases, we can consider it effortless and without latency for many users. We believe it Permissive-for-Unknown-Callers, because if an unknown caller cannot reach the recipient, it is presumably because the recipient did not authorize the call. Users are expected to assign a Secret or to stick with the Legacy mode when giving their number in real-life interactions. Otherwise, why would they give their number in the first place? Recipients may post their enrollment URL to web pages. Callers could share any collected Secrets with third parties. Technically, these scenarios allow an unknown caller to reach a recipient.

The Authorization Secret credentials can be preserved intact even if the caller or the recipient change their phone number. This is very advantageous compared to other techniques.

4.2. Deployability

The proposal is not Negligible-Changes-to-Infrastructure, although the changes to the protocols are minor adaptations in the worst case.

Therefore, we deem the impact of the changes smaller than, for instance, in Payment at Risk mechanisms, as for the inter-carrier relationships the business model is not substantially changed.

Deploying this proposal will require an effort by all TSPs worldwide, as they will have to do the necessary engineering to at least support the transmission of the Secret by callers. Smartphone recipients will have to update their devices and install apps. Callers will have to adapt as well, with varying degrees of effort. For example, PBXs (Private Branch eXchanges) have dial plans in their configuration for outgoing call routing, which may have to be adapted for long format number dialing.

The deployment of our proposal cannot be performed as standalone within a user or a TSP. For it to be effective, it requires a coordinated effort among multiple agents globally. There will presumably be a period of transition where companies will invest in engineering and users will spend time configuring the Authorization Secret feature, without the formers being able to offer an effective service. Users will not benefit from this feature until everybody else (or, we should say, the majority) has completed the migration process. Despite these migration efforts, it is likely that no hardware upgrades are necessary on the network infrastructure or the endpoints. If that is confirmed, this would make this proposal much more attractive.

It is not entirely Negligible-Changes-to-Call-Setups. However, as mentioned, a high volume of calls are made from contact agendas or databases, so this property would be satisfied for many users.

Regarding the Low-Resource-Requirement, it requires significant engineering efforts to set up the system, but the computational power requirements for establishing calls are low, and they are distributed among millions of mobile devices. Our proposal satisfies No-Third-Party-Involvement when Secrets are exchanged manually. As mentioned, setting up the infrastructure requires many parties involvement. Once it is set up, the involvement of the TSP is minimal. Therefore, it is Low-Maintenance and Negligible-Cost-per-Call.

4.3. Robustness

The proposal is Effective-Against-Dynamic-Caller-ID-Spoofing, Effective-Against-Unavailable-Caller-ID, Effective-Against-Multiple-Identities, Effective-Against-Answering-Machine-Detection and Effective-Against-Dynamic-Audio-Content. We highlight that this proposal does not rely on the received caller ID unless enabled by the user. Effectiveness is not affected by audio content in any way.

It is Effective-Against-Targeted-Caller-ID-Spoofing except when making use of caller ID whitelisting in Flexible mode of operation. However, these should be narrowed down scenarios. Unlike solutions targeting SPIT prevention, our authorization approach offers an effective method for a variety of scenarios of unwanted calls: robocalls in targeted commercial and political spam, fraud and scams, stalking, etc. It is adaptable to both VoIP and switched telephony technologies.

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

It is Deterministic. Some proposals rely on score weighting or Artificial Intelligence methods. Those may require a considerable work of parameter tuning, try-and-error, and progressive re-designing. They may also require more maintenance, as spammers find new ways to work around them. Non-deterministic approaches are generally more prone to false positives/negatives. In contrast, the current proposal relies on well-defined and predictable mechanisms.

4.4. Brute-Force Resilience

Mainly when targeting a specific recipient, brute-force techniques can be leveraged to attempt to hack a valid Secret. For POTS callers who have to dial Static Secrets, there is a trade-off between the added security obtained by incrementing the number of digits, and the usability of reducing them.

To make such attacks less attractive, we recommend that carriers charge a small amount of money for calls rejected by recipients reporting an Authorization Failed error code. This charge should not represent a big change in the business model of telephone providers, as it could be seen as the users having these calls diverted to some voicemail. There are exceptions, such as when the recipient effectively revokes the Authorization Secret, but does not want the caller to be charged or be aware of such revocation.

It should not be difficult for mobile devices to detect brute force attacks. Then, users may react, for example, by contacting the TSP to report the issue and trace back the attacker. See APPENDIX A for a mathematical analysis of the brute force strength in a realistic worst-case scenario.

4.5. Economy of mechanism

Despite the engineering efforts and adaptations necessary to deploy this proposal at the beginning, it follows the principle of economy of mechanism. At the end of the day, the recipient shares a sort of password with their contacts (but a different password for each of them), and the contacts have to provide the password when calling the recipient. The Keep It Simple, Stupid principle inspires this mechanism. We do not introduce new complex algorithms; instead, we build on top of proven algorithms. Adaptations need to be made over telephony protocols, as we described in section III. However, these adaptations are minor.

4.6. Context-awareness

The authorization given to someone for them to call us, is done precisely at a significant moment: when exchanging numbers with that person/entity. This moment is when the user had the context for deciding to allow the call back (which can be revoked anytime). Also, when saving an issued Authorization Secret, it can manually or automatically have attached metadata of related contact, timestamp, location, personal notes, or others. The system is based on independent and contextual decisions made by the user ahead of time, rather than on a complex system that tries to guess what the user wants. The issuance of Authorization Secret stems precisely from the real-world interactions that motivate such authorizations.

4.7. End-to-End, Decentralized, Accountable

Authentication systems based on digital certificates usually involve some central party to validate the identity of the users, as with Certificate Authorities. If a directory of public keys is used, it remains the question of who validates these keys and how. By providing end-to-end Authorization of calls, we give the the users the power to autonomously decide what calls they want to receive. No central authority issues or revokes authorizations or decides what calls are allowed and what calls are filtered, otherwise this could lead to false positives, false negatives, privacy issues, or censorship complaints. Note that when using web services, we are not delegating the authority over our Secrets. They just help with sharing these Secrets conveniently. Once the initial enrollment is made, they should not intervene any more, and they should not keep record of the Code/Seed of the Secret.

Regarding accountability, when a recipient finds that they got an unwanted call despite it having a valid Secret, they can easily trace back the initial real-world interaction where that Secret was issued. In this way, if someone shares Secrets with other people without the issuer permission, and these people call the recipient; the recipient can identify the offender and take appropriate action.

4.8. Anonymity

It must be made clear that this proposal is not for authentication but for authorization. The problem we are dealing with here is receiving unwanted calls. Nothing else. For this reason, we believe that the authorization property of our system fits with such a purpose. If we wanted to make sure that the caller is who they claim to be, then we would focus on authentication-based solutions. We do not necessarily have to migrate to an authentication-based solution to avoid unwanted calls. Anonymity is a property that can, if not should, be preserved. On the one side, using the Legacy mode plus the regular Caller ID removal, we allow for unidentified calls, which is fundamental in many public services. On the other side, we can benefit from using the Authorization Secret feature while still allowing for

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

unauthenticated calls. For example, a recipient may post an enrollment URL to a forum or chat, asking for anonymous inputs. Let's say they want to do a poll or flirt with people. The recipient can revoke the Secret in a single operation when no more inputs are desired, to avoid being bothered any more.

4.9. Confidentiality and Integrity

Providers and carriers involved in the transport of the call, and any applications used for this feature, will have visibility over the Secret. Therefore, governments should develop regulations to protect users, making companies treat Secrets as confidential information. Secrets should not be logged or monitored except if requested by the customer or if required to process the communication (for example, temporarily storing the Secret of a voice mail message or SMS in the message center). This inhibition of logging prevents leaking of sensitive information in case the provider suffers a data breach.

Using Static Codes opens the possibility of replay attacks, as mobile communications can be eavesdropped. If that happens, the user may revoke an issued Secret as soon as they get an unwanted call. If the Secret was linked to a phone contact number, they may just get in touch with that contact to give them a new Secret. If the problem persists, recipients may check their location at the moments when previous legitimate calls were received with that particular Secret, to enclose the location or path taken by a Stingray device.

We highly recommend having the mobile operating system natively perform all Secret-related functions so as to mitigate risks associated with trusting third party apps. Also, we recommend using audo-deploy only during the initial phase.

Any Authorization Secrets stored in the device (either issued or received) should be kept safe for the information at rest. To see the stored Secrets, the user should confirm authentication with the device password or with biometrics. The Secrets DB should be kept encrypted within the device. It should also be backed up regularly, as we commonly do with the contacts agenda. The backup would allow to restore information in case of damage, or to revoke issued/received Secrets in case of theft.

There is no guarantee that the Secret will remain unaltered in transit, either when provided or received. However, we have not identified any relevant motive for an attacker to corrupt the Secret number. A TSP might block a TOTP-enabled call before it is established and sell the Code to third parties for immediate use. Our system does not prevent that, but if the incident persists with different IDs and the phone does not detect brute-force, this might indicate a network-based attack or a compromise of the issued Secrets DB.

4.10. Multi staging

Our proposal is meant to be a first-level of defense against undesired calls, but there may be scenarios where further processing with another technique is needed. For example, when a Strict-mode recipient rejects a call because it lacks the Authorization Secret, it may be because the caller uses a device or a service provider that does not support that feature. To prevent blocking legitimate users, other SPIT detection techniques can be leveraged [9] [10]. Alternatively, voice recognition can be deployed to assist with the process. More precisely, if the voice is matched with the list of trusted voice profiles of the recipient, the call is accepted; otherwise the caller is asked to record their name and the recipient can decide to accept (and optionally whitelist the voice profile) or reject (and optionally blacklist the voice profile) the call. Those services could be offered by the TSP, possibly for a fee.

4.11. Downsides and limitations

We focused on mobile users with smartphone, but our model is extensible to other intelligent systems. Unfortunately, regular landline phones or basic cell phones lack the necessary capabilities as recipients. A possible alternative for landline recipients is to use a mobile app provided by the TSP as a mediator to authorize incoming calls. All household members would have their app integrated. For rotary phones, TSP app integration can be used for dialing, or the Secret can be provided verbally to a voice recognition system, or else the recipient should switch to Flexible/Legacy mode.

Wealthy motivated attackers may be able to brute force a targeted recipient. As a mitigation, we should either increase the cost of rejected calls, or increment the number of digits of the Secret. The latter would overflow ISDN's Keypad facility field limits, so it implies modifying the ISDN protocol.

If PBX recipients use this feature and ISDN is the access technology, the Authorization Secret feature could be misused as a sort of Direct Inward Dialing (DID). In other words, the recipient could use the Secret value to somehow extend the contracted numbering range with the TSP without actually having to pay for this extension. We can think of it as the Secret being used as the extension number in a company. As a workaround, TSPs may charge for ISDN channels and Secret support, rather than for DID.

Companies could use the Secret as a sort of user-tracking cookie. For example, the company's website might issue a different Secret on each web request and keep track of the originating IP address. When customers call, the company would know their IP address and potentially other metadata. If the user is

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

behind a privacy network the company would still know so, which is already some information. Even if everybody used privacy tools, this would be a means for the company to identify individuals across multiple calls. Public services and customer service centers should be bound to always provide phone numbers in Legacy mode.

In TOTP-based Codes there might be scenarios where the call setup takes long enough to refuse legitimate calls if using the typical 30-second timer values. To avoid this, we can adjust the window timer allowance. Especially for Static Codes, users might get a false sense of security. As discussed in section IV-I, it is feasible for motivated attackers to get a valid Static Code and spoof the caller ID. When a recipient gets a call with a valid Secret, they may be tempted to relax their security awareness. However, users should keep the same security best practises as if the Authorization Code feature never existed. This means that they should always apply zero trust, even if both the calling number and the Secret are valid.

5. Conclusion

We offered here a revised proposal to prevent unwanted calls, using state-of-the-art techniques to offer better properties. Required protocol adaptations are minor. Likely, no hardware upgrades should be necessary. Required software upgrades, engineering efforts and user configurations are expected to be deployed gradually. More research is needed, such as checking implementation feasibility in real equipment, deploying a proof of concept, or testing usability.

As a drastic solution to prevent SPIT, we could remove the public telephony infrastructure and use VoIP-based mobile app calls instead. However, this idea is easier said than done. Among other reasons, VoIP is not yet a reliable alternative for emergency calls. Also, telephony providers make profits with PSTN calls (including spam calls).

At this point, the reader might wonder what is the incentive for TSPs to consider any anti-spam technology at all. They could just let users live with it or let them adopt other already-available solutions that do not require network upgrades (despite them being inconvenient or less effective). But such a strategy might be no better than our proposal, as end users, frustrated, may end up getting tired of spam calls and abandon PSTN technologies completely.

6. References

[1] Contact Method. (2020). Federal Trade Commission, Fraud Reports.

[2] First Orion. (2018). Scam Call Trends and Projections Report.

[3] Truecaller. (2021). Insights US Spam and Scam Report.

[4] National Center for Health Statistics. (2021). Wireless Substitution: Early Release of Estimates from the National Health Interview Survey.

[5] Kimball, S. H., Levy, T., Venturelli, H., and Miller, S. (2014). “Interactive voice recognition communication in electoral politics: Exploratory metadata analysis,” American Behavioral Scientist, vol. 58(9) 1236-1245.

[6] Ono, K., and Schulzrinne, H. (2009). “Have I met you before? Using cross-media relations to reduce SPIT,” Proceedings of the 3rd International Conference on Principles, Systems and Applications of IP Telecommunications.

[7] Tu, H., Doupé, A., Zhao, Z., and Ahn, G. (2016). “SoK: Everyone hates robocalls: A survey of techniques against telephone spam,” IEEE Symposium on Security and Privacy.

[8] Saverio, N., (2006). “SPIT prevention: state of the art and research challenges.”, Proceedings of the 3rd Workshop on Securing Voice over IP.

[9] Quittek, J., Niccolini, S., Tartarelli, S., and Schlegel, R. (2008). “On Spam over Internet Telephony (SPIT) prevention,” IEEE Communications Magazine.

[10] Ajmal Azad, M., and Morla, R., (2011). “Multistage SPIT detection in transit VoIP,” International Conference on Software, Telecommunications and Computer Networks.

[11] M’Raihi et al., (2011). Internet Engineering Task Force Trust. Request for Comments: 6238.

[12] 3rd Generation Partnership Project. (2021). Specification 23040 Release 17, Technical Specification Group Core Network and Terminals, Technical realization of the Short Message Service (SMS).

[13] International Telecommunication Union. (1991). CCITT Q.767 Recommendation, Application of the ISDN user part of CCITT Signalling System No. 7 for international ISDN interconnections.

[14] International Telecommunication Union. (2010). E.164 Recommendation, The international public telecommunication numbering plan.

[15] International Telecommunication Union. (2018). Q.850 Recommendation, Usage of cause and location in the Digital Subscriber Signalling System

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

No. 1 and the Signalling System No. 7 ISDN user part.

[16] International Telecommunication Union. (1988). Q.23 Recommendation, Technical features of push-button telephone sets.

[17] Rosenberg et al., (2002). The Internet Society. Request for Comments: 3261.

[18] International Telecommunication Union. (1998). Q.931 Recommendation, ISDN user-network interface layer 3 specification for basic call control.

[19] 3rd Generation Partnership Project. (2021). Specification 24008 Release 17, Technical Specification Group Core Network and Terminals; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3.

Appendix A

Brute Force Analysis

We will here analyze a realistic worst case scenario of brute forcing of the Authorization Secret against a given subscriber who uses Strict mode and issues 20,000 valid Secrets. We want to know what d = minimum number of digits of the Secrets satisfies that the expected number of attempts an attacker has to do prior to finding a valid Secret is greater than 200,000,000. If we establish that rejected calls have a fixed cost of $0.01 per call, this should deter most attackers.

It is assumed that the attacker knows the subscriber phone number but has no prior knowledge about the issued Secrets. They may make use of unlimited memory space or other techniques to ensure not repeating already attempted Secrets. As a reminder, these Secrets are decimal, and generated randomly. Many users are expected to have a mix of static and TOTP Codes. However, we will consider that all Codes are static, which is an easier to hack scenario compared to the Codes with a variable TOTP component.

We model this problem as if all possible issuable Secrets were put in order in a roulette. The black slots represent the valid Secrets, whereas the white slots represent the invalid ones. We imagine the hacker using the roulette to get the first attempted Secret at random. Figure 5 shows an example of a space of 30 possible Secrets, of which 8 are valid. Let's suppose the roulette gives a 16. Therefore, the attacker first tries with that, which turns out to be unsuccessful. After this first attempt, in Figure 6 we virtually extract the black slots, and we leave them only as a mark. Subsequent Secrets are taken sequentially. Because the Secrets were issued at random by the user, the attacker scan is also random despite that we now take the slots sequentially. In the example, we now scan 17,18,19... We stop at 20, which is right before we cross a black slot. In this example we spend 5 attempts prior to finding a black slot, and the system is hacked in the 6th attempt. The average distance in between black slot marks is the number of white blocks over the number of black blocks. If we select a point at random in the circle, the expected distance to a black block is precisely half of that.

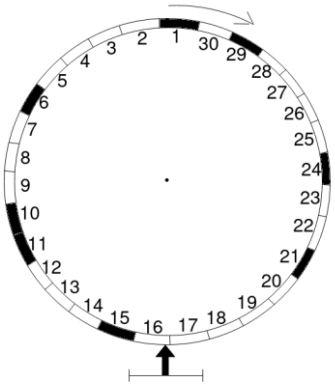

Figure 5. Roulette Model of the first Hacking attempt

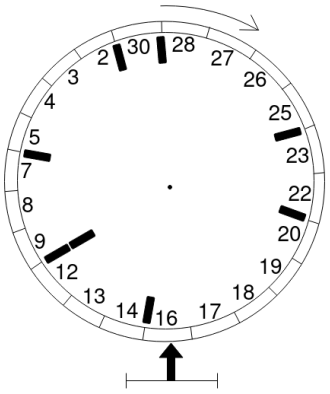

Figure 6. Roulette Model of the Subsequent Hacking attempts

While keeping this model in mind, we calculate the expected value, in terms of d, that satisfies our requirements.

$$E(\text{\# attempts}) > 2 \cdot 10^8$$

$$0 \cdot P(\text{hack at 1st}) +$$

$$E(\text{\# attempts when } \neg \text{ hack at 1st}) \cdot P(\neg \text{ hack at 1st}) > 2 \cdot 10^8$$

$$\frac{1}{2} \cdot \frac{10^d - 2 \cdot 10^4}{2 \cdot 10^4} \cdot \frac{10^d - 2 \cdot 10^4}{10^d} > 2 \cdot 10^8$$

$$L = \frac{(10^d - 2 \cdot 10^4)^2}{8 \cdot 10^{d+12}} > 1$$

We know that d⩾5. In TABLE II we plug in values onto d to get L and solve the inequality.

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118

Table 2. Plug in values for d

d	***L***
5	0.00008...
6	0.0003...
7	0.001...
8	0.003...
9	0.01...
10	0.03...
11	0.1...
12	0.3...
13	1.1...

Therefore, $d = 13$.

Copyright © 2022, Infonomics Society | DOI: 10.20533/ijisr.2042.4639.2022.0118