Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 14, 2014

E. Rescorla
RTFM, Inc.

July 13, 2013

Secure Caller-ID Fallback Mode
draft-rescorla-stir-fallback-00

Abstract

A major challenge with RFC 4474-style identity assertions has been that SIP operates in highly mediated and interworked environments. SIP requests may pass through gateways, policy enforcement devices or other entities that receive SIP requests and effectively act as user agents, re-initiating a request. In these circumstances, intermediaries may recreate the fields protected by the RFC4474 signature, making end-to end integrity impossible. This document describes a mechanism for two compliant endpoints to exchange authentication data even in the face of intermediaries which remove all additional call signaling meta-data or which translate from SIP into protocols incapable of understanding identity meta-data (e.g., where one side is the PSTN).

material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 14, 2014.

Copyright Notice

Table of Contents

1. Introduction 4
2. Operating Environment 4
3. Architectural Options 5
4. Strawman Architecture 6
4.1. Phone Number Authentication 6
4.2. Call Placement Service 6
4.3. Security Analysis 7
4.3.1. Substitution Attacks 8
5. Some Potential Enhancements 9
5.1. Encrypted CPRs 9
5.2. Signed CPRs 9
5.3. Credential Lookup 10
5.4. Federated Verification Services 10
5.5. Escalation to VoIP 10
6. Security Considerations 11
Appendix A. Acknowledgements 11
Author's Address 11

1. Introduction

A natural design for providing caller authentication is to attach a signature to the call setup messages (e.g., a SIP INVITE). This is incompatible with much of the existing communications environment. Most calls from telephone numbers still traverse the PSTN at some point. Broadly, these calls fall into one of three categories:

- One or both of the endpoints is actually a PSTN endpoint.
- Both of the endpoints are non-PSTN (SIP, Jingle, ...) but the call transits the PSTN at some point.
- Non-PSTN calls which do not transit the PSTN at all.

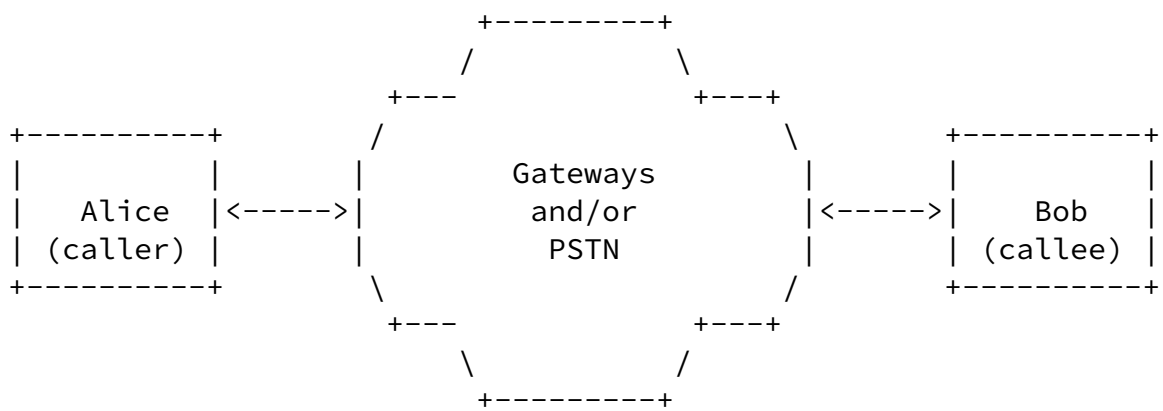
The first two categories represent the vast majority of these calls. The network elements that operate the PSTN are legacy devices that are unlikely to change at this point. However, these devices are also unlikely to pass signatures--or indeed any inband signaling data--intact. In many cases they will strip the signatures; in others, they will damage them to the point where they cannot be verified. In either case, any in-band authentication scheme does not seem practical in the current environment.

While the core network of the PSTN remains fixed, the endpoints of the telephone network are becoming increasingly programmable and sophisticated. Landline "plain old telephone service" deployments, especially in the developed world, are shrinking, and increasingly being replaced by three classes of intelligent devices: smart phones, IP PBXs, and terminal adapters. All three are general purpose computers, and typically all three have Internet access as

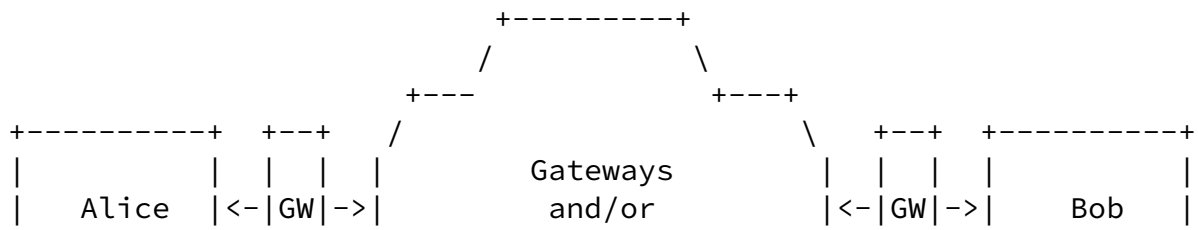
well as access to the PSTN. This provides a potential avenue for building an authentication system that changes only the endpoints while leaving the PSTN intact.

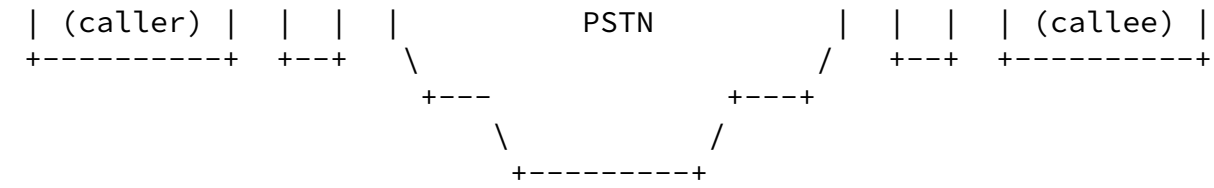
2. Operating Environment

This section describes the environment in which the proposed mechanism is intended to operate. In the simplest setting, Alice is calling Bob through some set of gateways and/or the PSTN. Both Alice and Bob have smart devices which we can modify, but they do not have a clear connection between them: Alice cannot inject any data into the system which Bob can read, with the exception of her asserted E.164 number. Thus, this number is the only value which can be used for coordination.



In a more complicated setting, Alice and/or Bob may not have a programmable device, but have a programmable gateway that services them, as shown below:





In such a case, Alice might have an analog connection to her gateway/ switch which is responsible for her identity. Similarly, the gateway would verify Alice's identity, generate the right caller-id information and provide caller-id information to Bob using ordinary POTS mechanisms.

3. Architectural Options

Because endpoints cannot communicate directly, any solution must involve some rendezvous mechanism to allow endpoints to communicate. We call this rendezvous service a "call placement service" (CPS). In principle they could communicate any information, but minimally we expect it to include a "call placement record" (CPR) that describes the caller, callee, and the time of the call. The callee can use the existence of a CPR for a given incoming call as rough validation of the asserted origin of that call. (See Section 6 for limitations of this design.)

There are roughly two plausible dataflow architectures for the CPS:

- The callee registers with the CPS. When the caller wishes to place a call to the callee, it sends the CPR to the CPS which forwards it to the callee.
- The caller stores the CPR with the CPS at the time of call placement. When the callee receives the call, it contacts the CPS and retrieves the CDR.

While the first architecture is roughly isomorphic to current VoIP protocols, it shares their drawbacks. Specifically, the callee must maintain a full-time connection to the CPS to serve as a notification channel. This comes with the usual networking costs to the callee and is especially problemtatic for mobile endpoints. Thus, we focus on the second architecture in which the PSTN incoming call serves as the notification channel and the callee can then contact the CPS to retrieve the CPR.

4. Strawman Architecture

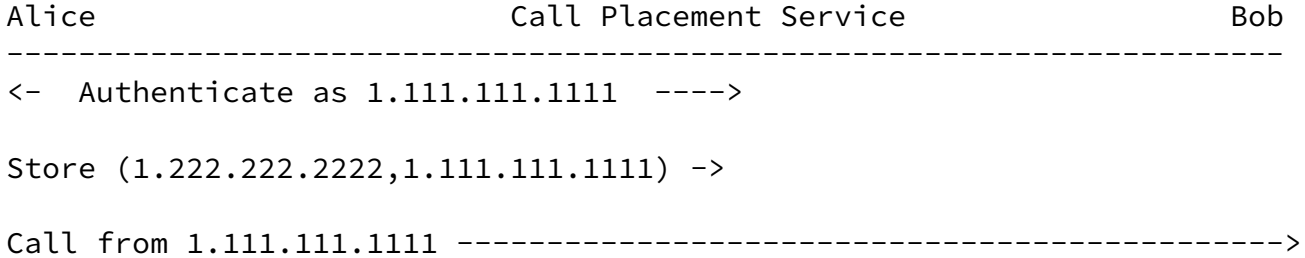
In this section, we discuss a strawman architecture along the lines described in the previous section. This discussion is deliberately sketchy, focusing on broad concepts and skipping over details. The intent here is merely to provide a rough concept, not a complete solution.

4.1. Phone Number Authentication

We start from the premise that each phone number in the system is associated with a set of credentials which can be used to prove ownership of that number. For purposes of exposition we will assume that ownership is associated with the endpoint (e.g., a smartphone) but it might well be associated with a gateway acting for the endpoint instead. It might be the case that multiple entities are able to act for a given number, provided that they have the appropriate authority. The question of how an entity is determined to have control of a given number is out of scope for this document.

4.2. Call Placement Service

An overview of the basic calling and verification process is shown below. In this diagram, we assume that Alice has the number +1.111.111.1111 and Bob has the number +2.222.222.2222.



<- Authenticate as 1.222.222.2222 ---->

<-------------- Retrieve call record
from 1.111.111.1111?

(1.222.222.2222,1.111.111.1111) -->

[Ring phone with callerid
= 1.111.111.1111]

When Alice wishes to make a call to Bob, she contacts the CPS and authenticates to prove her ownership of her E.164 number. Once she has authenticated, she then stores a Call Placement Record (CPR) on the CPS. The CPR should also have some sort of timestamp to prevent replay. The CPR is stored under Alice's number.

Once Alice has stored the CPR, she then places the call to Bob as usual. At this point, Bob's phone would usually ring and display Alice's number (+1.111.111.1111), which is provided by the usual caller-id mechanisms (i.e., the CIN field of the IAM). Instead, Bob's phone transparently contacts the CPS and requests any current CPRs from Alice. The CPS responds with any such CPRs (assuming they exists). If such a CPR exists, he can then present the callerid information as valid. Otherwise, the call is unverifiable. Note that this does not necessarily mean that the call is bogus; because we expect incremental deployment many legitimate calls will be unverifiable

4.3. Security Analysis

The primary attack we seek to prevent is an attacker convincing the callee that a given call is from some other caller C. There are two scenarios to be concerned with:

- The attacker wishes to simulate a call when none exists.
- The attacker wishes to substitute himself for an existing call as described in Section 4.3.1

If an attacker can inject fake CPRs into the CPS or in the

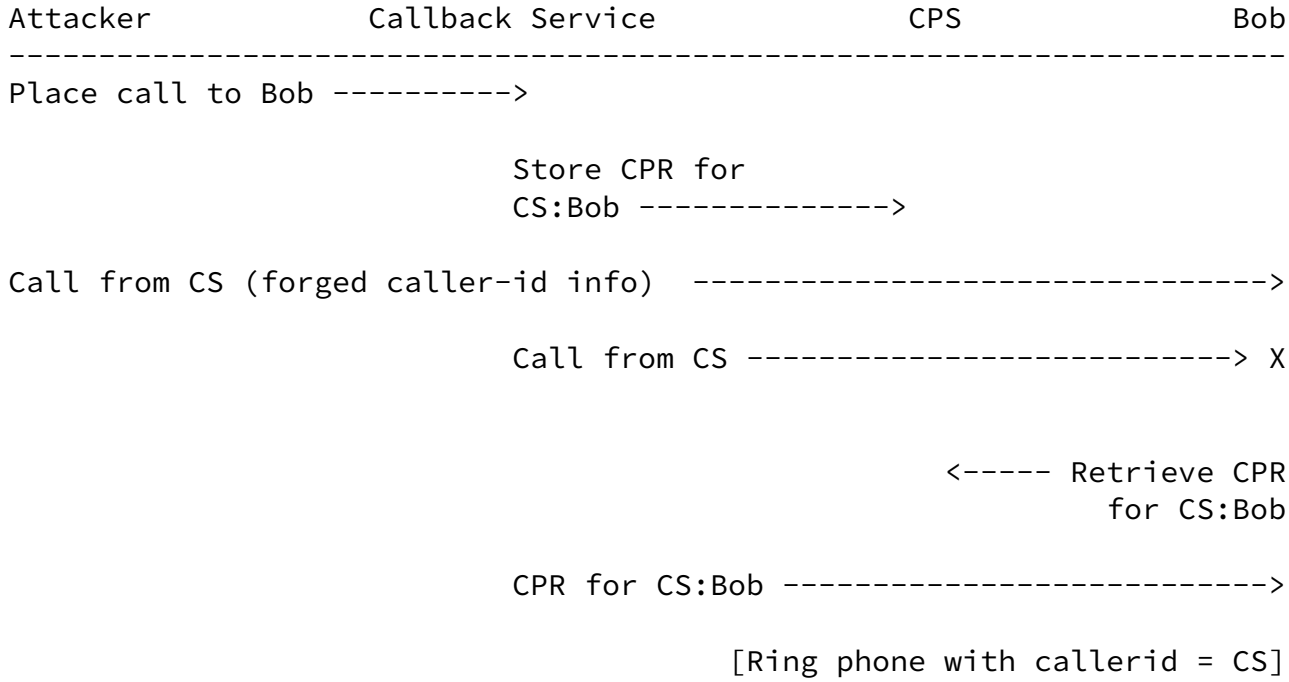
communication from the CPS to the callee, he can mount either attack.

In order to prevent this, either the communication to the CPS should be secured in transport (e.g., with TLS) or the CPRs should be digitally signed by the caller and verified by the callee (Section 5.2. For privacy and robustness reasons, both are preferable. In particular, if only transport security is used, then a compromised CPS can forge call origination information.

The entire system depends on the security of the authentication infrastructure. If the authentication credentials for a given number are compromised, then an attacker can impersonate calls from that number.

4.3.1. Substitution Attacks

All that receipt of the CPR proves is that Alice is trying to call Bob (or at least was as of very recently). It does not prove that this particular incoming call is from Alice. Consider the scenario in which we have a service which provides an automatic callback to a user-provided number. In that case, the attacker can arrange for a false caller-id value, as shown below:



In order to mount this attack, the attacker contacts the Callback Service (CS) and provides it with Bob's number. This causes the CS to initiate a call to Bob. As before, the CS contacts the CPS to insert an appropriate CPR and then initiates a call to Bob. Because it is a valid CS injecting the CPR, none of the security checks mentioned above help. However, the attacker simultaneously initiates a call to Bob using forged caller-id information corresponding to the

CS. If he wins the race with the CS, then Bob's phone will attempt to verify the attacker's call (and succeed since they are indistinguishable) and the CS's call will go to busy/voice mail/call waiting. Note: in a SIP environment, the callee might notice that there were multiple INVITEs and thus detect this attack.

5. Some Potential Enhancements

Section 4 provides a broad sketch of an approach. In this section, we consider some potential enhancements. Readers can feel free to skip this section, as it is not necessary to get the flavor of the document.

5.1. Encrypted CPRs

In the system described in Section 4, the CPS learns the CPRs for every call, which is undesirable from a privacy perspective. The situation can be improved by having the caller store encrypted CPRs. A number of schemes are possible, but for concreteness we sketch one possibility.

The general idea is that each user's credentials are not just suitable for authentication to the CPS but also are an asymmetric key pair suitable for use in an encryption mode. When Alice wants to store a CPR for Bob she retrieves Bob's credentials (see Section 5.3) and then encrypts the CPR under Bob's public key. [The encryption needs to be done in such a way that if you don't have Bob's key, the message is indistinguishable from random. This is straightforward, but not compatible with typical secure message formats, which tend to indicate the recipient's identity.] The CPR is then stored with the CPS under Alice's identity. When Bob receives a call, he just asks the CPR (anonymously) for any calls from Alice to anyone. He then trial-decrypts each and if any of them is for him, he proceeds as before. In this way, the CPR learns Alice's call velocity but not who she is calling.

5.2. Signed CPRs

In the system described in Section 4, the CPS can forge CPRs. This threat can be removed by having the CPR signed by the originator along with a timestamp. If such a signature is required, the originator cannot make bogus calls appear to be valid but can still make valid calls appear to be bogus by removing the relevant CPRs.

5.3. Credential Lookup

In order to encrypt the CPR, the caller needs access to the callee's credentials (specifically the public key). This requires some sort of directory/lookup system. This document does not specify any particular scheme, but a list of requirements would be something like:

Obviously, if there is a single central database and the caller and callee each contact it in real time to determine the other's credentials, then this represents a real privacy risk, as the central database learns about each call. A number of mechanisms are potentially available to mitigate this:

- Have endpoints pre-fetch credentials for potential counterparties (e.g., their address book or the entire database).
- Have caching servers in the user's network that proxy their fetches and thus conceal the relationship between the user and the credentials they are fetching.

Clearly, there is a privacy/timeliness tradeoff in that getting really up-to-date knowledge about credential validity requires contacting the credential directory in real-time (e.g., via OCSP). This is somewhat mitigated for the caller's credentials in that he can get short-term credentials right before placing a call which only reveals his calling rate, but not who he is calling. Alternately, the CPS can verify the caller's credentials via OCSP, though of course this requires the callee to trust the CPS's verification. This approach does not work as well for the callee's credentials, but the risk there is more modest since an attacker would need to both have the callee's credentials and regularly poll the database for every potential caller.

We consider the exact best point in the tradeoff space to be an open issue.

5.4. Federated Verification Services

The discussion above is written in terms of a single CPS, but this

potentially has scaling problems, as well as allowing the CPS to learn about every call. These issues can be alleviated by having a federated CPS. If a credential lookup service is already available, the CPS location can also be stored in the callee's credentials.

5.5. Escalation to VoIP

If the call is to be carried over the PSTN, then the security properties described above are about the best we can do. However, if

Alice and Bob are both VoIP capable, then there is an opportunity to provide a higher quality of service and security. The basic idea is that the CPR contains rendezvous information for Alice (e.g., Alice's SIP URI). Once Bob has verified Alice's CPR, he can initiate a VoIP connection directly to Alice, thus bypassing the PSTN. Mechanisms of this type are out of scope of this document.

6. Security Considerations

This entire document is about security, but the detailed security properties depend on having a single concrete scheme to analyze.

Appendix A. Acknowledgements

Jon Peterson provided some of the text in this document. The ideas in this document come out of discussions with Richard Barnes, Cullen Jennings, and Jon Peterson.

Author's Address

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
USA

Phone: +1 650 678 2350
Email: ekr@rtfm.com