An Affordable Solution for Authenticated Communications for Enterprise and Personal Use

Matthew Chiang  
Department of Computer Science  
Georgetown University  
37th & O St, NW  
Washington, DC 20057 USA  
Email: mlc248@georgetown.edu

Eric Burger  
Department of Computer Science  
Georgetown University  
37th & O St, NW  
Washington, DC 20057 USA  
Email: eric.burger@georgetown.edu

Abstract—The top complaint received by both the U.S. Federal Trade Commission and Federal Communications Commission is illegal robocalling. One of the issues enabling illegal robocalling is the relative ease of a bad actor to spoof, or lie, about who is calling. Specifically, the bad actor can set their Caller ID, or the number that is displayed when one receives a phone call, to whatever they want. The Internet Engineering Task Force (IETF), the SIP Forum, and the Alliance for Telecommunications Industry Solutions (ATIS) have been jointly working on a solution, called STIR, to provide authentication services for the asserted caller ID. However, the focus of that work, particularly by the joint ATIS / SIP Forum Network-Network Interconnect Work Group, has been on service providers doing the authentication and verification. This article describes an implementation of the STIR technology targeted for small enterprise and consumer use. Our implementation is the first to be suitable for the enterprise environment. It is extremely lightweight and affordable, while still providing necessary security and integrity. We deployed our implementation in an enterprise environment, building upon the open source Kamailio SIP server.

Index Terms—SIP; PSTN; Communications Software; Security, Trust and Privacy; Systems and Software Engineering; Cyber Security; Robocalling

I. INTRODUCTION AND PROBLEM STATEMENT

The motivation behind this project is the volume and consequences of spoofed calls and illegal robocalls plaguing consumers of telecommunication—both over IP networks using the Session Initiation Protocol (SIP) and the legacy public switched telephone network (PSTN). Calls that originate in the IP domain can be spoofed relatively easily due to a lack of authentication, and this vulnerability can be propagated into the PSTN domain. These calls cause several real problems, including illegal and unwanted telemarketer calls; people pretending to be government agencies and demanding "overdue payments"; SWATting, where a criminal targets another for assassination or arrest by placing a 911 call with the target’s phone number as the Caller ID, usually with a report of an armed and dangerous situation underway, resulting in the police responding in force; and so on. Furthermore, in the US, illegal robocalls are the #1 complaint at both the Federal Communications Commission (FCC) and Federal Trade Commission (FTC). RFC 7340 [1] gives a more detailed description of the problem.

The IETF STIR approach is to have some entity cryptographically sign the signaling, specifically the caller’s identity, so that a recipient can trust that the asserted identity is correct and not a forgery [2]. Because most recipients of illegal robocalls are on the PSTN or wireless mobile PSTN-interconnected networks, the focus of development has been on service providers implementing this technology. Because of their business relationship and connection characteristics, the service provider should know the identity of the caller. For example, in the wireline PSTN, the physical copper cable between the caller and the telephone company central office identifies the phone line, and thus telephone number, of the caller. In the wireless mobile case, the handset performs a cryptographic login, using either credentials on a SIM card or a phone’s MEID, to the network before the user can place calls. In the broadband integrated telephony case, as opposed to over-the-top voice, the service provider knows the cable modem’s identity and thus the caller’s number.

In the service provider-verified scenario, where the originating service provider knows who the caller is by virtue of the caller using the service provider’s directly connected telephony service, the service provider will sign the SIP signaling, adding an Identity header that includes the information being signed, including the Caller ID, and the signature itself. Signing is performed by a distinct
authentication server.

Complementary to the authentication server is a verification server. Again, with a focus on service provider implementations, the terminating service provider will validate the integrity of the asserted Caller ID by validating the signature in the Identity header. As well, the terminating service provider will implement a set of policies, such as only trusting signatures that use certificates issued by a limited set of well-known certificate authorities. This model of service providers trusting only service providers, and not providing actual end-to-end authentication is emblematic of service provider-centric approaches such as [3].

One expectation is that as STIR becomes more deployed, if one receives a call that is not signed, it may be blocked by the network or end user device as being untrustworthy. As such, there is a need for the average phone user, particularly an enterprise user, to want to both prevent spoofed calls and have their legitimate calls delivered. The work described in this paper delivers on that need.

The following five sections are as follows. The background section provides a brief overview of the relevant SIP headers and the approaches to date for providing authenticated identity in SIP, almost all of which are exclusively targeted to service provider networks. The following section describes our novel architecture and implementation, targeted to enterprise networks. We follow this with a section on our implementation, which we then follow by an analysis of our implementation compared to prior approaches. Finally, we describe some avenues for future research.

II. BACKGROUND

A. SIP

Session Initiation Protocol (SIP) [4] is a signaling protocol for initiating multimedia sessions over IP, such as text messaging, voice, and video. A SIP protocol message consists of several headers, including To, From, Date, Call-ID, CSeq, Content-Type, Content-Length. Content-Type and Content-Length refer to the body of the message, which is typically Session Description Protocol (SDP), which describes the media. Of note for identity, the To field describes the intended recipient of the call, the From field describes the identity of the caller, and the Date field describes the time the caller placed the call.

Base SIP suffers from a lack of authentication for caller identity, as anyone can put whatever they want into the headers. Thus, an attacker can modify the From header to impersonate another party.

B. SIP Identity

SIP Identity [5] extended base SIP by providing authentication of SIP messages via two additional SIP headers: Identity and Identity Info. The Identity header contained a hash and signature over several fields, including To, From, call-id, method, CSeq, Date, and Body. The Identity Info field included a URI to access the certificate to verify the signature. The protocol was never widely distributed due to its restrictions. For example, it didn’t allow intermediary SIP servers to alter headers or bodies because the signatures were taken over so many fields. Other problems with SIP Identity are identified in [6].

C. SHAKEN

SHAKEN [7], as developed by ATIS and the SIP Forum, is a service provider based framework for authenticated identity. The SHAKEN framework uses the STIR Authenticated Identity protocol in a way that is tailored towards carriers. Beyond the normal specification of STIR Authenticated Identity, SHAKEN details ways in which service providers deploy the protocol. Among these includes handling which incoming requests to accept as authenticated. In SHAKEN, any signature by a carrier is accepted. Thus, the responsibility of authentication is placed solely on the service providers.

D. P-Asserted-Identity

P-Asserted-Identity [8] adds a P-Asserted-Identity header field to the SIP message. P-headers are private headers, meant for conveying information within a private network. In practice, these are service provider networks. A network element in the service provider network inserts the P-Asserted-Identity header with the identity of the caller. By configuration and by virtue of running on a closed, service provider network, users cannot change the P-Asserted-Identity header. However, P-Asserted-Identity is not reliable for inter-carrier operations, unless it is somehow signed.

E. Other Approaches and Issues

Verification Involving PSTN Reachability (VIPR) [9] is a protocol attempting to solve the issue of inter-domain SIP routing. It does this by creating a ticket that is sent along with a call in a SIP header. The ticket causes a handshake that authenticates the caller. However, this can only occur after a PSTN call occurs between the caller and receiver. Thus, the first call is not authenticated.

Some have proposed using something akin to symmetric encryption for the signaling. For example in [3], the scheme encrypts the digest of the message, then creates a randomly generated session key, then encrypts the SIP message, the caller’s signature, and the session key. Like the SIP Authenticated Identity solution, this is inflexible, requires a long key exchange, and requires that each party
have the other party’s key ahead of time. However, for general public network communication, where anyone can call anyone, it is not realistic to expect everyone to have everyone else’s key.

We took a software analog to the approach [10] took for providing security in TETRA radios. That approach was to find a low-cost, easy to implement circuit to warn users of attacks. However, that model requires additional hardware, which we aim to avoid.

III. STIR AUTHENTICATED IDENTITY FOR ENTERPRISE DEPLOYMENTS

STIR Authenticated Identity [2] ensures caller authentication through an Identity SIP header, which is a signed hash over at least the To, From, and Date fields. SHAKEWEN specifies the Persona Assertion Token (PASSporT), which describes the signature itself.

A. Authentication Service

The authentication service is at the sender’s end and adds an Identity header to outgoing messages. The process we use is as follows.

1) Check that the server has the authority to authenticate originator
2) Authenticate the originator
3) Verify that the date is valid
4) Calculate signature
5) Add signature and URI indicating certificate location to Identity header
6) Pass on message to the next hop

B. Verification Service

The verification service sits on the receiving end and verifies an incoming message containing an Identity header is valid. The process is as follows.

1) Determine the originator’s identity (telephone number or URI)
2) Identify credential for validation (obtain certificate)
3) Check for freshness of date (max of 60 seconds between receive time and time sent)
4) Confirm that the given Identity header is valid using the certificate
5) If valid, send to end user

C. Security

STIR’s Authenticated Identity cryptographically assures caller authentication. It is an improvement over SIP Authenticated Identity in that it allows network devices to change or add other fields while retaining the essential To and From fields. Session Border Controllers and other kinds middleboxes often perform such modifications to a SIP message. As such, even though no one was modifying the From and To fields, by being over protective and signing too many fields, SIP Authenticated Identity was not deployable in the real world.

IV. IMPLEMENTATION

Our implementation builds upon the open-source SIP server Kamailio [11]. We added the STIR Authenticated Identity protocol as a module. Since both the Kamailio server and our module are available as open-source software, it is readily available for use.

A. Hardware

We built and deployed the STIR-enabled Kamailio servers on Raspberry Pi 3 devices, with a 16 GB flash drive. We chose this device as it is representative of an embedded controller appropriate for enterprise or personal use.

B. Software

After downloading and installing the Kamailio SIP server, we wrote the authentication module in C.

Unlike the carrier implementation built elsewhere, our module integrates the signing and verification servers. Both servers are implemented together in one program, but the program can gauge its current role from:

- If a SIP message is received with the Sender being an authorized user, then it acts as the authorization server. It will sign the message and pass it on.
- If a SIP message is received with the Receiver being an authorized user, then it acts as the verification server. It will verify the Identity header, and if valid, pass it on to the Receiver.

We took the cryptography library from OpenSSL, implementing the baseline STIR protocol requirements of ECDSA with the P-256 curve and SHA-256 hashing algorithm.

Additionally, we modified the SIP message parser to parse the STIR Identity header in the message.

C. Testing

The module was tested by running two instances of the Kamailio server and two instances of SIP user agents (UA). One UA was connected to one machine, designated the “Sender”, and the other UA to the “Receiver” machine. Blink [12] was used as the SIP client. The call flow is shown in Figure 1.
In our implementation, Bob takes the role of the sender and Alice of the receiver. A call is initiated from Bob to Alice, through the authentication and verification proxies. A Wireshark analysis verifies that there is a correct Identity header in the SIP Invite sent.

Then, the roles were reversed and the test repeated.

Calls with missing or incorrectly-formed Identity headers would result in the verification service dropping the call. These cases were tested and confirmed.

V. ANALYSIS

All previous implementations of STIR’s Authenticated Identity protocol are geared towards service provider use. This project is unique in that it is geared towards enterprise and personal use. As such, it has clear benefits for the consumer, including end-to-end authentication, lightweight cost, and affordability.

A. End-to-End Authentication

Our project uses a different trust model from the service provider model. In the service provider model, only the authority and signatures of the service provider is trusted, instead of the end-user. The service provider, in turn, trusts the end-user’s identity via its proprietary connection or network sign-in. This model is centralized with the service provider as the authority.

In our model, the process is democratized. Here, there is true end-to-end authentication between the users. The service provider’s intervention is unnecessary, provided that we have a certificate that the receiving party can trust. This is beneficial in that users do not need to rely solely on the service provider for authentication. It is possible that the service provider is untrustworthy or becomes compromised; in these cases, the service providers could produce false authentication or verification and the end-users would be left vulnerable. In our model, if the end-users can securely exchange a certificate or sufficiently prove ownership of the URI of a certificate, they can communicate with authentication.

B. Architectural Complexity

Our implementation is lightweight compared to the service provider model, providing authentication with a minimal amount of setup and servers. The SHAKEN framework, shown in Figure 2, which is intended for large-scale service provider deployment of STIR’s Authenticated Identity, contains several parts.

Note that our architecture, as shown in Figure 1 above, needs only the authentication service, verification service, and a way for the verification service to access the certificate remotely. Many of the extra network components in the SHAKEN model are unnecessary for enterprise use—for example, because of manageable amounts of traffic, we store keys and perform signatures directly on the authentication server rather than retrieving it via HTTPS from another server. A similar optimization is used for the verification service. Note that the HTTPS request to obtain the certificate is still required in our framework, so the verification service can verify the signature. Our implementation is intended to be lightweight, requiring only necessary components for an enterprise deployment.

C. Affordability

Our implementation used Raspberry Pi 3’s and, for further validation, on Virtual Machines with equivalent memory. Since the system performed well on a Raspberry Pi, and likely on even simpler hardware, we believe that it should be possible for consumers to use our solution for call
authentication—both in terms of financial cost and computer resources. Furthermore, all software is open-source. The bottom line: we built a complete STIR-compliant SIP server for under US$70 in a package under 50cc’s of volume.

VI. FUTURE RESEARCH

A. Hardware Security Module

In our implementation, the private key, which is used for signing the Identity header, is stored directly on the flash drive of the authentication and verification servers, in plaintext. This has obvious issues for security and integrity. A path for further research is to use a hardware security module (HSM) in order to sign the Identity header of SIP messages without the risk of storing the key in plaintext. An example of this approach is in [13]. However, this approach is not necessarily appropriate for personal- or enterprise-use as most HSMs tend to be bulky and expensive. In order to keep the project inexpensive, one could do something similar to that described by [14] and [15].

B. Interoperability with SHAKEN/STIR

We believe that our solution is fully interoperable with the carrier-oriented SHAKEN/STIR approach described above. The industry has created a STIR interoperability test bed. As of the time of writing of this paper that testing is underway.

C. Out of Band STIR

Out of Band STIR (OOB STIR) [16] is a protocol currently being developed in the IETF that aims to bring authentication to calls which travel through the PSTN for endpoints that do have Internet connectivity. This protocol assumes a lack of end-to-end SIP connectivity. In OOB STIR, the caller only has the phone number of the called party. The protocol assumption is that each end-user has an IP connection to some global service, but not to each other or SIP. The proposed solution revolves around two global services to store and relay PASSporTs and public keys.

D. Interoperability with the PSTN

A related project currently in the works is ensuring authentication of SIP calls where the media traverses the PSTN. STIR’s Authenticated Identity protocol only works for end-to-end SIP calls, but in many cases, it is impossible to have end-to-end SIP; that is, the call traverses the PSTN at some point during routing. In particular, we are focusing on calls that are SIP to SIP with enough IP connectivity to exchange SIP Invites and tokens, but not enough IP connectivity to carry the audio of the call—thus, the audio must be transported over the PSTN. Unlike with SIP, the PSTN cannot simply add an additional header containing a signature. We are exploring additional protocols to ensure authentication of caller identity in these cases.

VII. CONCLUSION

We developed an implementation of authenticated communication over IP that is affordable and practical for enterprise and personal use. It is different from the service provider model as specified by SHAKEN in that our implementation does not rely on the service provider for authentication and is more lightweight and simpler, all while retaining the same cryptographic security. Whereas SHAKEN relies on carriers to provide authentication and verification, our implementation places these at the endpoints, allowing for end-to-end authentication between the caller and callee.

VIII. Acknowledgements

The authors would like to acknowledge the thoughtful comments of the reviewers of the paper. We would also like to thank Mr. Chris Wendt of Comcast Corporation for his sharing of the SHAKEN diagrams. Finally, we acknowledge the support of the affiliates of the Security and Software Engineering Research Center (S2ERC) at Georgetown University, including support from the National Science Foundation under grant IIP-1132046.

REFERENCES
