1789

GEORGETOWN UNIVERSITY

S2ERC

SECURITY AND SOFTWARE
ENGINEERING RESEARCH CENTER

S^{2}ERC Project: Next Generation Caller Identification
Authors: Dr. Eric Burger and Ms. Julia Kieserman
Date: 23 June 2016

Abstract

This report, a result of the *Next Generation Caller Identification* project in the PSTN Transition program in the S^{2}ERC, examines a proposal for asserting caller identity in the all-IP telecommunications network, 4474bis in conjunction with the work being progressed in the ATIS SIP Forum IP NNI Task Group.

STIR is the standard developed by IETF that defines a signature to verify the calling number, and specifies how it will be transported in SIP "on the wire" whereas SHAKEN is the framework document developed by ATIS/SIP Forum IP-NNI task force to provide an implementation profile for service providers implementing STIR. The objective of SHAKEN is to provide guidance to implementers to ensure interoperability. The mechanism passes a JWT token (JSON Web Token) in the SIP INVITE request in the Identity header. It also integrated the IETF passport mechanism, which specifies a token format for authenticating sent information. This report describes the mechanism and how it addresses the caller identification challenges faced by telecommunications users.

This material is based upon work supported by the National Science Foundation under Grant No. 1362046 and the industry affiliates of the Security and Software Engineering Research Center (S^{2}ERC). The views and analysis provided are entirely our own and not attributable to any other party. Support for this work includes funding from the S^{2}ERC affiliates iconectiv and Comcast. Payments are made to Georgetown University and the funds are used to cover the expenses of the study and related academic and research activities of the institution.

Summary

The Internet Engineering Task Force (IETF)[1] is a global standards development organization focused on the creation and maintenance of protocols that create and work over the Internet.

The IETF developed the principal interoperable Voice over Internet Protocol (VoIP) standard, the Session Initiation Protocol, or SIP. The IETF identifies standards by RFC numbers. For example, RFC 3261 defines the current version of SIP.

RFC 4474, *Enhancements for Authenticated Identity Management in the Session Initiation Protocol*, described a method for providing cryptographic assurance that, amongst other things, the calling party identifier asserted in the call request has not been changed in transit. For a number of technical reasons, RFC 4474 was not functional. Moreover, RFC 4474 only assured no one altered the caller's identity claim in transit. Even so, a caller that lies about its identity could still lie about its identity.

Secure Telephone Identity Revisited (STIR)[2] is an IETF work group focusing on delivering a secure identity attestation mechanism for SIP calls. The attestation mechanism means that if the originator of a call lies about their identity, regulatory agencies or others could positively identify who made the claim. Moreover, the mechanism allows third parties, such as trusted carriers, to make the attestation of the identity of the caller. The work group adopted formatting and protocol mechanisms from work done in the ATIS/SIP Forum NNI task force.[3]

The goal of STIR is to enhance the SIP protocol to provide a mechanism for an authoritative identity provider to assert the originator of a SIP call is likely to be who they assert they are. The goal of these enhancements is to make it considerably more difficult for bad actors to spoof the identity of a call for malevolent or other purposes. Examples of such activities are spoofing voice messaging or credit card validation services to get access to the victim's voice messages or credit, respectively; spoofing the identify of a victim for assassination by making threatening calls to public safety officials (a.k.a., SWATting); engaging in confidence schemes by masquerading as legitimate enterprises looking for information or cash (e.g., banks for personal identification or the IRS for swindling); or to get through blocked caller lists (e.g., robocalling).

STIR mitigates these problems, but is not designed to provide a 100% solution. Moreover, STIR does not work in all call scenarios. The following table is a summary of the call scenarios and STIR's contribution to the mitigation of the caller ID spoofing problem.

In short, STIR addresses asserted identities in the U.S. domestic SIP-signaled network. The protocol mechanism is Internet-wide and as such international. However,

[1] https://www.ietf.org/

[2] https://datatracker.ietf.org/wg/stir/charter/

[3] http://www.atis.org/01_strat_init/IP-NNI/index.asp

the specifics of how the authorization mechanism works, for all practical purposes, will be a U.S.-centric solution for the foreseeable future. This is because the approach requires one to trust the certificate authority issuing the signing credentials. That is an issue of policy, not technology. As such, that will require industry agreement or regulation, which by its nature will be dealt with by the various national telecommunications authorities.

Table 1 - STIR Calling Scenarios

Originating Network	Terminating Network	Mitigation of Spoofing
PSTN	PSTN	No impact
SIP-Domestic	SIP-Domestic	Significant impact
SIP-Domestic	PSTN	Potential impact
PSTN	SIP-Domestic	No impact
SIP-International	PSTN	No impact
SIP-International	SIP-Domestic	Little impact

From a technology perspective, there is no way of using the mechanism in a reliable manner for PSTN-originated calls. A PSTN-originated call would be coming from a legacy class 5 or PBX switch to a media gateway. There is no technical way of proving the provenance of the asserted caller ID or billing number (ANI) of a call in the PSTN. That is why this issue continues to exist. There could be administrative ways of improving the provenance of the asserted caller ID coming from the PSTN, such as if the access carrier will vouch for the veracity of the caller ID. This would be possible for those carriers who would be willing to verify that calls originating in their network have valid caller IDs. This is an issue for further study.

As such, the mechanism does offer potential to mitigate some PSTN-originated calls that have SIP gateways ‘close’ to the origination or where the trunk sources can be identified. In addition, international gateways could mark calls with numbers being asserted as a calling party number from the North American Numbering Plan (NANP), but originating from outside the NANP, as suspect.

STIR

Certificate Generation

Before the exchange of SIP messages, the authentication service creates a private/public key-pair by sending a CSR (certificate signing request) to a CA (Certificate Authority). After the CA confirms the authentication service is who it claims to be, possibly by some form of digest authentication, the CA sends the authentication service a signed certificate for signing SIP messages. This process is shown in Figure 1. Note that typically the authentication service will ask for a single key for signing multiple requests.

The authentication service uses this certificate for signing outbound requests. It is an implementation detail whether an authentication service will use a single or multiple keys. They may use multiple keys if, for example, the authentication service is providing different levels of attestation for different numbers. The authentication service may also use unique keys for every telephone number (TN) that it will sign for.

Certificate policy is a significant area for further work. We will address this issue later in the paper. For the foreseeable future, it is our expectation that each authentication service instance will use its own key for all of the numbers under its jurisdiction.

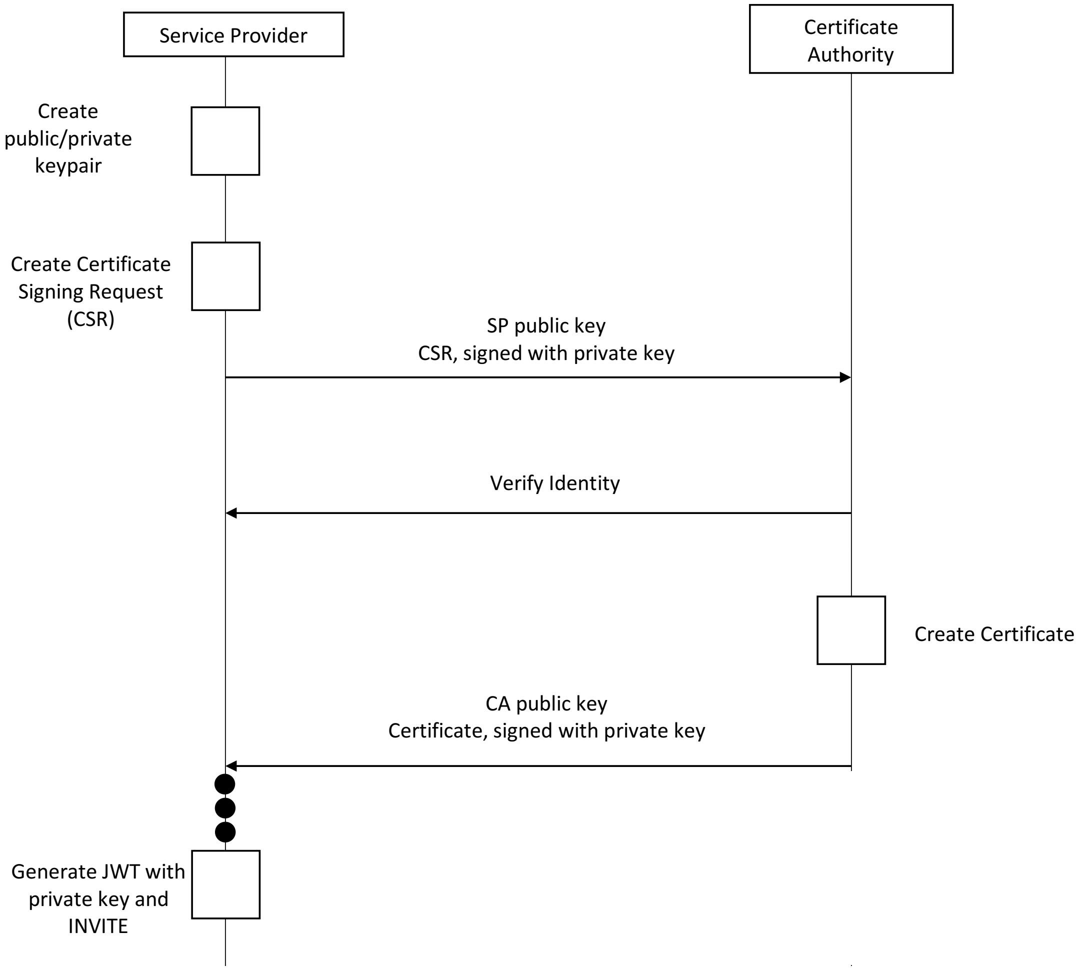
Service Provider
Certificate Authority
Create public/private keypair
Create Certificate Signing Request (CSR)
SP public key
CSR, signed with private key
Verify Identity
Create Certificate
CA public key
Certificate, signed with private key
Generate JWT with private key and INVITE

Figure 1 - Certificate Generation

Protocol Overview

Figure **2** shows the functional components of STIR. The SIP UAC (user agent client) places calls as it always did. Specifically, it forms INVITE and MESSAGE method messages, and sends those methods to the authentication service. The authentication service contains the private key of the service provider, as well as a certificate for authentication.

Note that the authentication service can be the service provider itself, which we expect for large carriers, or it could be a hosted service, which we expect for many small carriers.

Before authenticating the messages, the authentication service must first validate both that it is responsible for the identity in the From header of the INVITE (referred to as the identity field) and that the SIP UAC can validly claim that identity. Let us examine some typical scenarios.

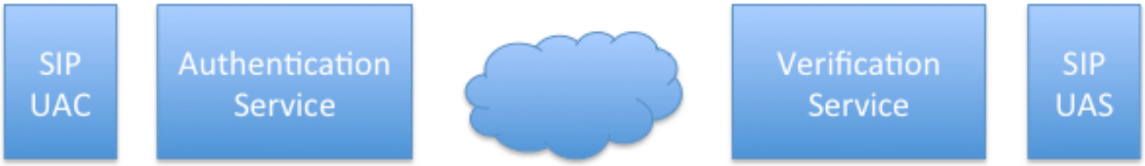

Figure 2 - STIR Functions

For landline SIP origination, there will most likely be an analog telephone adapter (ATA) originating the SIP call. Presumably, as a service provider is handling the call, there has been some sort of cryptographic authentication between the ATA and the SIP registrar. As such, when the INVITE arrives from the ATA, the service provider knows the identity of the caller, and as such can sign the attestation. A similar process works for DOCSIS adapters and VoLTE devices. Note that for the VoLTE and DOCSIS scenarios, this attestation depends on the registration of the device with the particular service provider. It is up to the service provider to ensure that INVITE requests from their customer's network or, in the case of a VoLTE phone, have the appropriate From identity. As many VoLTE and IMS service providers are re-writing the INVITE message to enforce, for example, SDP limitations, the service provider, knowing precisely which access device or customer network originates the call can force the From field to reflect the actual identity of the caller.

STIR's primary use is for validating identities that claim E.164 numbers. If the identity field contains a SIP URI, the authentication service extracts the hostname component of the identity field and confirms the authentication service is in fact responsible for that domain and if the username corresponds to an E.164 number, possibly after translations. If the identity field uses the TEL URI scheme, the policy of the authentication service is used to determine if it responsible for the extracted identity. The authentication service may only proceed after confirming it is in fact responsible for the identity. After doing so, the authentication service authenticates the sender of the message, in one of several ways. The authentication service should also ensure that the Date header in the message is relatively accurate. As we will see below, this is to mitigate cut-and-paste attacks.

After validating the identity of the message, the STIR authentication function forms a JWS (JSON Web Signature) identity signature and adds an identity header to the request containing this signature.

The authentication service creates a JWS token with the following three components: a Javascript Object Signing and Encryption (JOSE) header, JWS payload and JWS signature. The payload contains the following elements of the SIP message: the quoted identity (either an address of record or telephone number in the From field), the quoted target (either an address of record or telephone number from the To field), and an encoding of the value of the SIP Date header field. The JOSE header defines the type and encryption algorithm used in the token. The signature is created as specified by JWS with the private key corresponding to the X.509 key certificate referenced in the header. The signature is base64 encoded and inserted into the identity header of the SIP Invite.

The authentication service then forwards the message normally towards the UAS, with the JWS signature as the value of the Identity header. Figure 3 shows the entire call flow, below.

A function on the receiving side is the verification service. The verification service receives the SIP message and verifies the signature passes. It is a matter of local policy what to do if there is no signature or if the signature verification fails. Presuming the SIP message passes verification, the verification service sends the message to the SIP UAS (user agent server) for processing.

Just as service providers can run their own authentication service, they may also choose to outsource the verification service to a third party.

Figure 3 shows the entire call flow of the request. The “Confirm authoritative over domain/number” step would be a typical digest challenge. However, for ATA and VoLTE scenarios, the access network could enforce identity management and make that step unnecessary. The term “A-(JWT)” refers to the constructed identity token, the signature over the From, To, and Date fields. ‘A+’ refers to the public key for the authentication service.

This new process is an updated version of RFC 4474. The changes to RFC 4474 include reducing the scope of the Identity signature, specifically removing CSeq, Call-ID, Contact, and the message body. This is because session border controllers (SBCs) often modify or destroy these headers. As such, a UAS or inbound proxy could never verify the delivered identity. The Identity-info header was also removed and combined with the identity header, and a new baseline signature algorithm was chosen to use RSA-SHA1.[4] STIR uses a generalized credential mechanism as described above, making it an independent process. As well, RFC 4474bis alters the identity-digest-string format for compatibility with JWT.

For simplicity, since the receiving end, whether terminating native SIP to the endpoint, SIP to an enterprise PBX, or through a SIP gateway to the PSTN, sees the same signaling, we will walk through the various origination scenarios (PSTN, SIP, enterprise) first and then walk through the various termination scenarios (PSTN or SIP).

Let us look at a number of deployment models, specifically a call from a native SIP UAC to a native SIP UAS, a PSTN call to a native SIP UAS, and a PSTN call to a PSTN termination. For all of these call flows, we will look at inter-carrier calls. For calls

[4] At the time of this writing it seems likely the IETF will converge on the stronger, elliptic curve cryptography algorithm suite.

within a carrier, the carrier could choose to implement STIR or use internal measures to verify the authenticity of calling data.

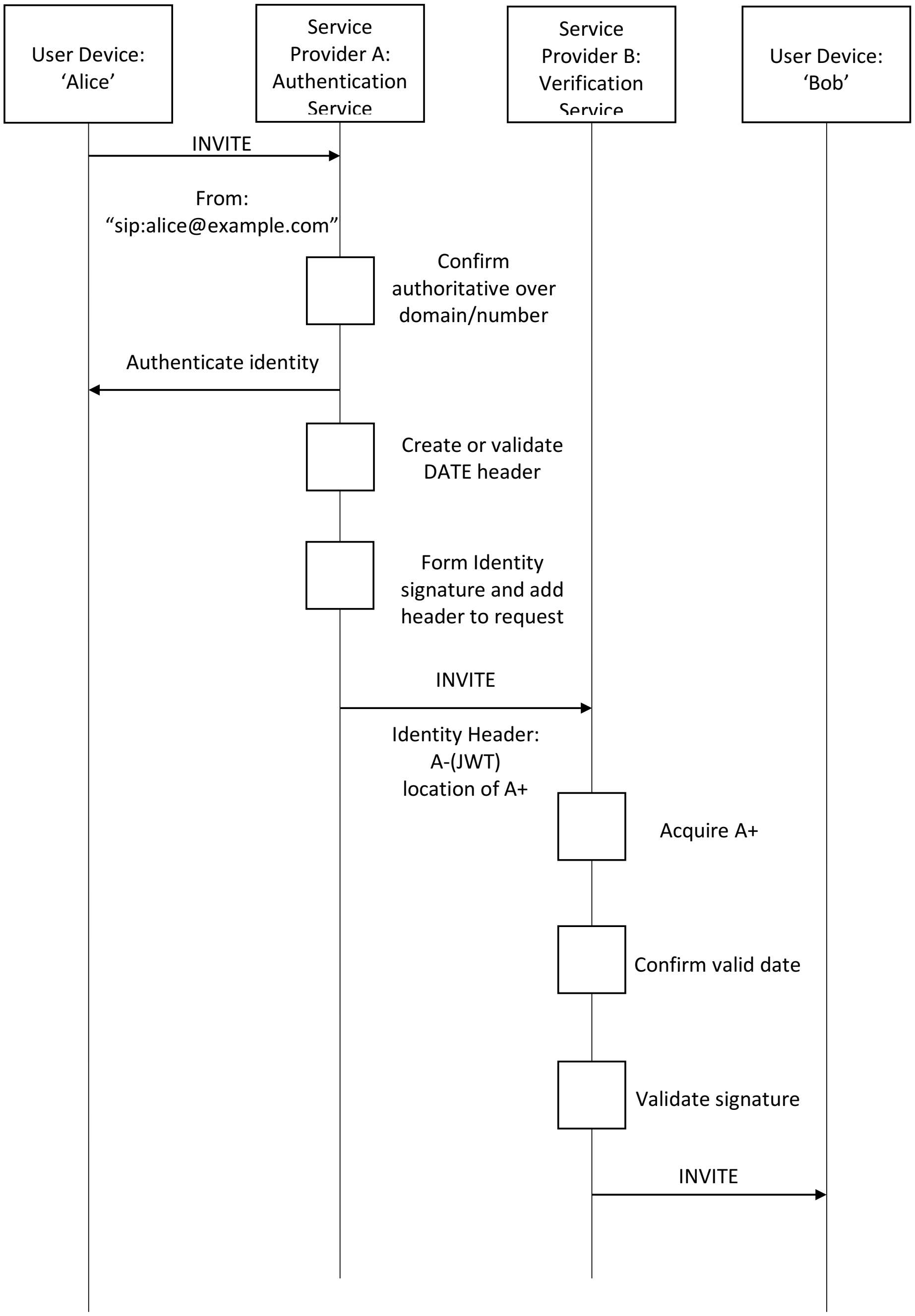
User Device: 'Alice'
Service Provider A: Authentication Service
Service Provider B: Verification Service
User Device: 'Bob'
INVITE
From: "sip:alice@example.com"
Confirm authoritative over domain/number
Authenticate identity
Create or validate DATE header
Form Identity signature and add header to request
INVITE
Identity Header: A-(JWT) location of A+
Acquire A+
Confirm valid date
Validate signature
INVITE

Figure 3 - Call Flow

Call Origination

Figure 4 describes a PSTN-originated call. In this case, the carrier is a legacy carrier with PSTN services. A call goes through the legacy network and ultimately through a media gateway into the carrier's IP network. The authentication service is invoked, possibly by a CSCF or on an egress SBC. Another model (not shown) is an integrated media gateway or a media gateway controller signs the message.

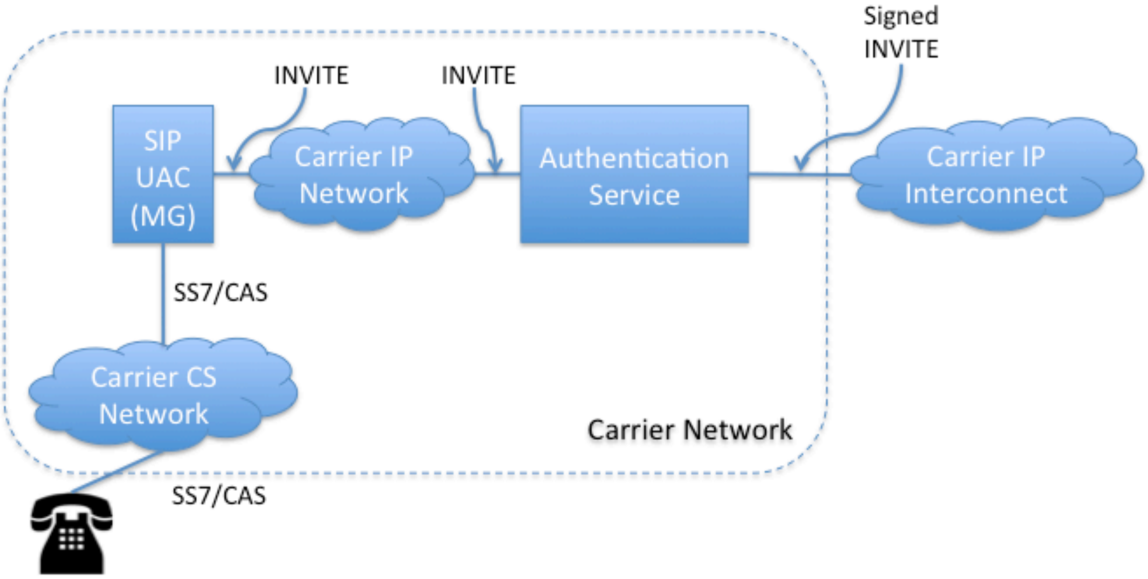

Figure 4 - Large Carrier Deployment Scenario: Origination and Signing

Figure 5 shows a possible deployment scenario for enterprises. The SIP Phone (UAC) [is supposed to] already does some sort of authentication with the outbound proxy, most likely through the REGISTER method. The expectation is the service provider will have tables, provisioning, or accept TN assertions present in the From field of future requests based on what the service provider's signing policy is. As such, the outbound proxy will be confident to sign messages on behalf of the phone.

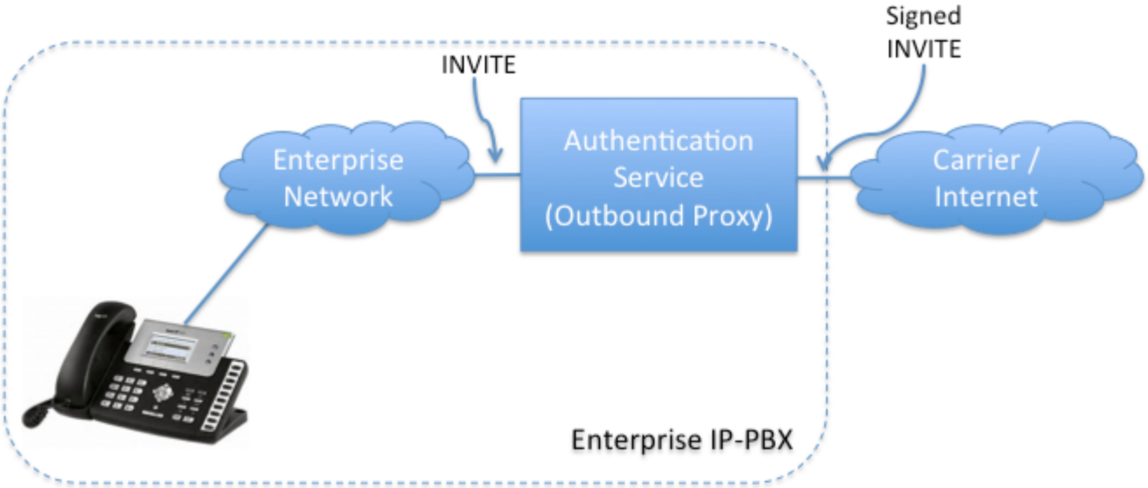

Figure 5 - IP PBX Enterprise Deployment Scenario

Note that the carrier could perform the authentication service at its ingress CSCF or SBC, signing the signaling on behalf of the enterprise and vouching for the authenticity of the From field.

Legitimate Spoofing

One common scenario for enterprises is the legitimate desire to spoof a callback number. For example, a particular individual customer service representative (CSR) may have their own direct dial number, yet the enterprise would like the enterprise's toll-free number to show up as the Caller ID and call-back number when the CSR calls a customer. As such, the enterprise wants to use a different Caller ID for the call than what it normally would be.

In standard SIP, this is straightforward. Since the From field identifies the *logical* entity placing the call, and the enterprise presumably allows the CSR to have the customer service *role*, the CSR can presumably REGISTER as a customer service entity, that is, they REGISTER with a From field with the enterprise's toll-free number. This works because the Contact field is what identifies the particular SIP user agent and is distinct from the From field, for just this purpose.

In any event, an authentication service can use any means to decide the policy for signing a message. For example, an enterprise that performs fundraising for Consumers' Union may notify their service provider they will be placing calls on behalf of Consumers' Union. Then, when the enterprise places calls with From fields indicating TNs belonging to Consumers' Union, the service provider will sign them. A particular twist here is it is unlikely for the enterprise to have the same service provider as Consumers' Union. However, the service provider presumably has some policy to believe the enterprise, perhaps because Consumers' Union vouches for the transaction.

With cryptographically signed messages, it is easy to discern and heavily sanction bad actors. An enterprise lying about its right to use someone else's number will quickly come back to the enterprise. Unlike the PSTN, even if the Caller ID of a call points to Consumers' Union, the cryptographic signature points to the originating service provider, who will have records of the calls made by the enterprise. Likewise, a service provider that regularly admits traffic with false From fields can be appropriately sanctioned.

Call Verification

Now let us look at the verification side of the equation.

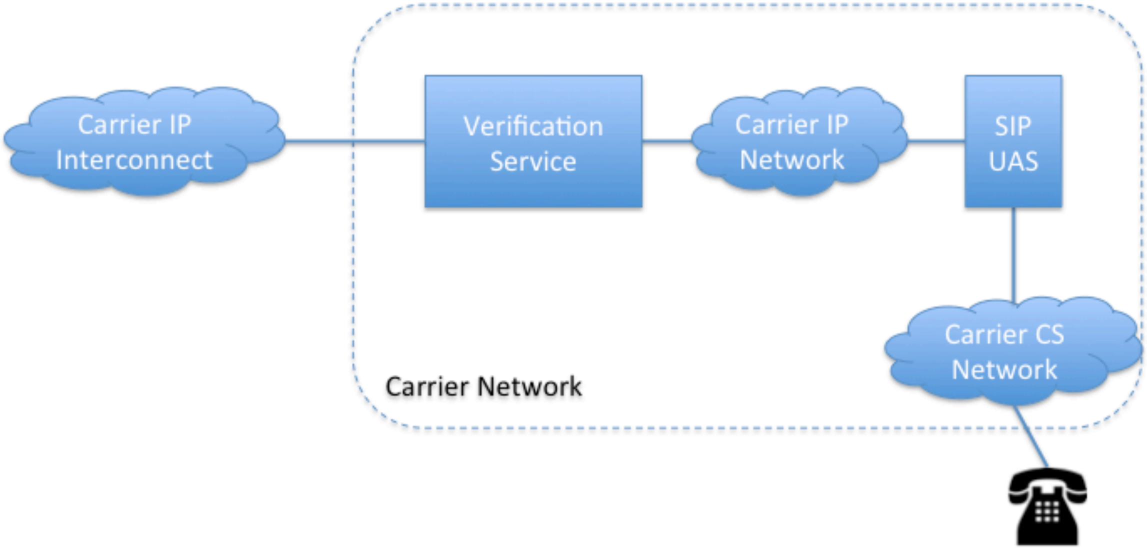

Figure 6 - Large Carrier Validation Service

Figure 6 shows an exemplary deployment of a verification service for a large carrier. On ingress, the verification service, typically an SBC or other CSCF, will verify the signature over the From and Date fields is correct. In order to verify the certificate, the verification service will need the sender's public key. It will use the public key in the certificate, accessible at the URI given in the 'x5u' parameter of the JOSE header in the identity signature.

Figure 7 is a ladder diagram depicting the call flow. Note that nothing changes for the UAC if the access carrier's CSCF, SBC, or enterprise IP PBX is providing outbound proxy (really B2BUA) services. Likewise, the UAC could have the Authentication Service integrated. As such, the original message from the integrated UAC would be the message with the Identity header.

The story of Figure 7 on the reception side is similar. The Verification Service, if an ingress CSCF, SBC, or IP PBX, would do the verification calculation. Depicted in Figure 7 is an example where the From and other protected fields pass verification.

What happens if the INVITE to the terminating carrier does not have an identity header? Figure 8 shows one possibility. Here the carrier still delivers the call to the UAS, but it modifies the From field to indicate the value may not be accurate. This is one of many possibilities, based on local policy. Another policy might be to black-hole the INVITE. In this figure, as in all the examples, the presumption is the Originating and Terminating service providers are different administrative domains. Yet another policy-based action the validation service can take if the validation fails is return a 438 response code. Figure 9 depicts this situation.

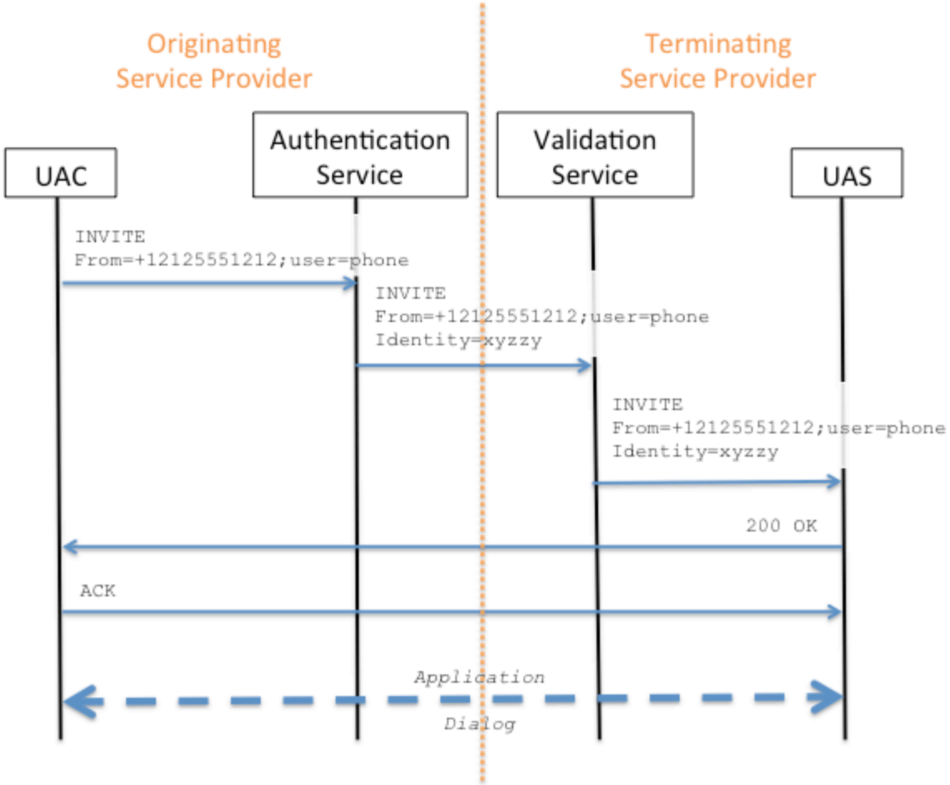

Figure 7 - STIR Ladder Diagram – Happy Path

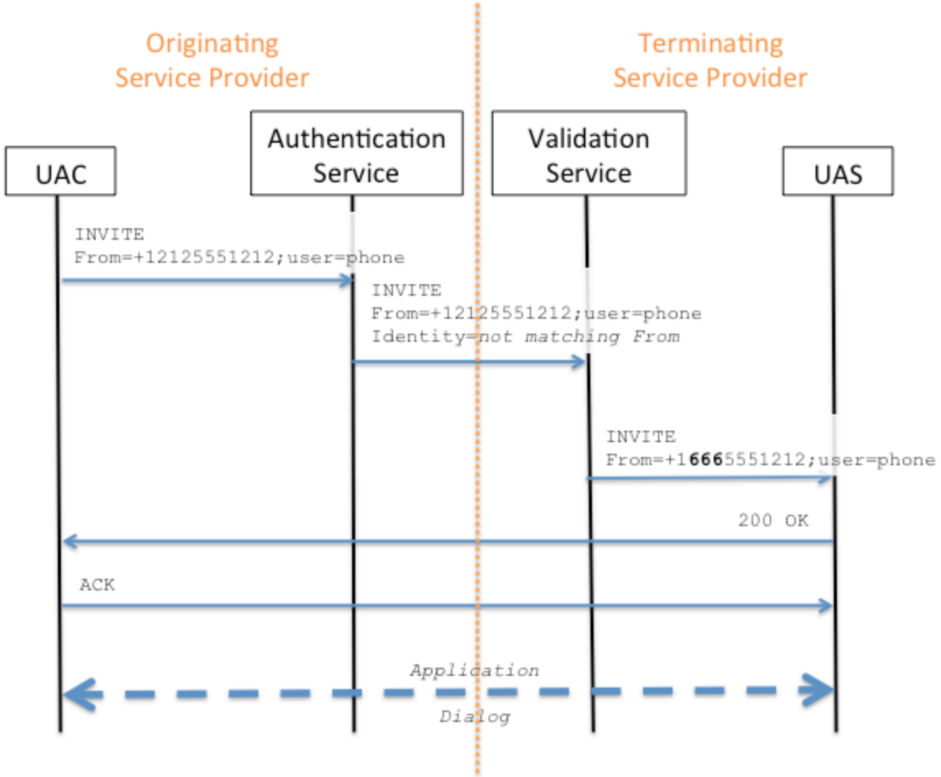

Figure 8 - Message Fails Verification: Session Continues

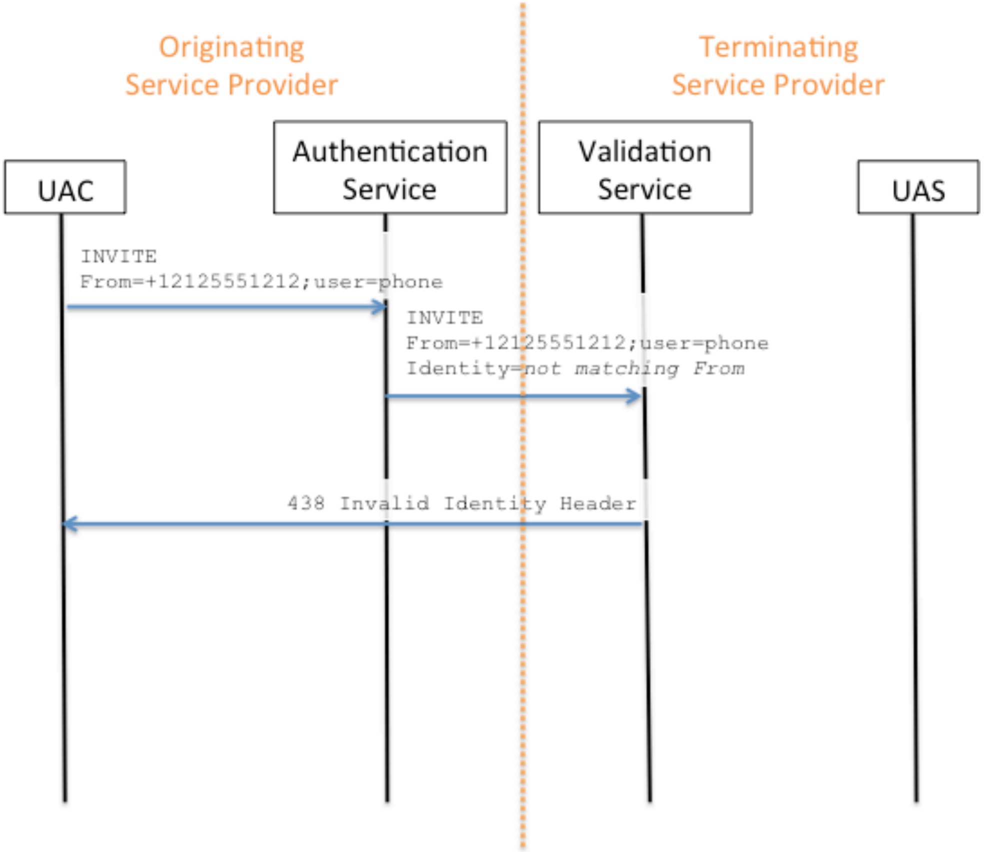

Figure 9 - Message Fails Verification: Session Rejected

There are some other cases. For example, there needs to be some international agreement as to roots and scope of authority. For example, the root for signing Vodafone UK +44 numbers is most likely the same root for signing BT UK +44 numbers. A logical entity for a UK root CA would be Ofcom. However, Germany is not likely to consider having Ofcom be the root for Vodafone's German +49 numbers. This becomes important for a situation such as when a U.S. carrier receives an INVITE, with a legal signature authenticating +1 512 416 7750 (the IRS help desk), with a validated chain of trust, with Nigeria as the root. That is not likely to be a real, valid call, whereas one signed by Verizon with the FCC, Symantec, or iconectiv as the root is likely to be real. The choice of which root CA to require is based on policy.

Since this is the realm of policy, it is out of scope for discussion in the IETF. However, solutions here will drive some of the technology boundaries on STIR.

Certificate Management

The controversies related to certificates are multifold. There is a lot of discussion about whether the certificates need to make an assertion over the telephone number itself or just to make an assertion about the authentication service. Simply put, the question is, is the certificate over the telephone number in the From header or is it an attestation by an authenticator (a.k.a. a service provider) that it vouches the From field is believable? In practice, what this comes down to is this: does the certificate represent the right to claim the TN being presented? Or, does the certificate represent who is producing the attestation being presented is allowed.

In the current protocol, certificates are issued specifically to the authentication service, so they are making an assertion about the authentication service (a.k.a a service provider) rather than the telephone number itself. However, the discussion is by no means over, and so below we discuss both possibilities.

Let us suppose the attestation is over the service provider. There are at least three perceived problems here. The first problem is that only calls that a trusted entity processes by can have a signature. This is a manifestation of the fact that presumably validation services will only trust messages signed by trusted entities. Cryptographically, we know the signer is a trusted entity because they are signing messages using a certificate with a signature from the trusted root CA.

The second perceived problem is this approach only provides cryptographic proof that the From field was not altered in transit. The only cryptographic proof about the validity of the value of the From field comes about because the verification service trusts the authentication service by virtue of the authentication service using a certificate issued by a trusted root CA. That is, the From field is accurate because the signer says so.

In reality, for the US market, trusting the signing authority is sufficient. Regulators can enforce the accuracy of the From field. Presumably the penalties for a service provider signing an erroneous From field will be high.

The third perceived problem is how to deal with STIR-signed, international messages. The validation service’s job appears easier if the certificate includes a TN or range of TNs that the From field asserts. If there is a match, great! Unfortunately, that is not sufficient. The validation service still needs to check to see of the root CA for the authentication service is trusted. Otherwise, anyone can go out, generate a certificate, and start signing phone calls.

What this requires is the validation service to know appropriate root CAs for various countries. This is not unreasonable. There are a limited number of countries, under 200. Moreover, there are already databases of carriers in all countries.

Let us suppose the attestation is over the number. This addresses the perceived cryptographic shortcoming of the above method of only knowing who signed the message. Specifically, having the attestation include the TN itself, which STIR requires no matter what the provenance of the certificate doing the signing, means the recipient can validate the number in the From field is cryptographically tied to the TN presented in the From field. This opens a number of very difficult issues.

First and foremost, the authority issuing the certificate to the signing entity has to sign that certificate in such a way that it vouches for the numbers being signed for. This is a non-trivial problem, given that the top two service providers each have hundreds of millions of numbers under management. Putting that in perspective, according to Verisign, .COM has just over 120,000,000 domains registered.[5] Moreover, that number is quite higher than the number of second-level domains that might have a certificate issued. For example, there are over 1,000,000 domains issued but not activated and, more importantly, squatters own a large percentage of the active domains. In other words, each of the two largest service providers have almost three orders of magnitude more active telephone numbers more than there are active .COM registrations.

As well, while we think of the Internet as a dynamic place, it does not compare with the PSTN. The top six TLD's combined (.COM, .NET, .ORG, .INFO, .BIZ, and .US) have, on a daily basis, about 160,000 adds, 120,000 deletions, and 190,000 transfers, for a total of just under half a million changes per day.[6] Compare this with the NPAC, which handles "tens of millions of transactions per day."[7]

If the certificate does make an assertion over the telephone number, is that assertion one telephone number per certificate or multiple numbers per certificate? What do we do when one ports a number? What happens if the certificate is over a range of TNs and a TN in the middle is ported and the current service provider is no longer authoritative for that TN? What happens when the number goes out of service? What happens if the root or authentication certificate authority is compromised and new certificates need to be generated for every TN?

One key characteristic of per-TN certificate management is, in fact, captured by the motivation for STIR in the first place:[8]

```
   [...] few SIP user
   agents today support the end-user certificates necessary to
   authenticate themselves (via S/MIME, for example)
```

In short, if we could do per-TN certificates, we would just use S/MIME, or S/MIME proxies. There would be no need for STIR. One could validate the integrity of the SIP message using 2002-era technology as described by RFC 3261 and simply mandate the certificate includes an attestation for the TN in question. Clearly, the community knows this is not practical.

Another issue is the governance of the public key infrastructure (PKI). Will there be a single, global root? A single root per country or region? A set of roots as we have for HTTPS in the Web?

[5] Verisign, *Zone Files For Top-Level Domains (TLDs)*, http://www.verisign.com/en_US/channel-resources/domain-registry-products/zone-file/index.xhtml, retrieved October 13, 2015. The count on that day was 120,374,019.

[6] DomainTools, *DailyChanges*, http://www.dailychanges.com, retrieved October 14, 2015. The numbers for that day were 159,218 adds, 116,077 deletes, and 183,357 trasnfers.

[7] Neustar, *Response to SB1335-01-Q-0740* (NTIA), Redacted Version, p. P-7.

[8] Peterson et al., *Authenticated Identity Management in the Session Initiation Protocol (SIP)*, draft-ietf-stir-rfc4474bis-06, October 19, 2015, work in progress.

One proposal is for there to be a single root CA, like we have for the DNS, most likely administered by the ITU-T. More likely would be one or a small number of per-country root CAs. Given 193 or so countries, this is not a huge burden. In the U.S., we would expect the FCC to designate one or more CAs, such as Symantec or iconectiv. Alternatively, the FCC could ask the industry itself, perhaps though an organization like ATIS, to contract out to a CA provider. Finally, we could use the Web model of 200+ global CAs, but that has obvious, well-known security holes, the most glaring being the Nigerian signing of an IRS phone number described above.

From one perspective, it makes sense for the authority handling number assignments to handle second-level CA assignment. They know who has the right to vouch for numbers. The good news is that since we are proposing standard, X.509 certificates, we can chose any of these CA models and the protocol just works. Which root CAs one should trust is an issue of policy and configuration. For example, whether a carrier issues per-TN certificates for its subscribers, uses a single certificate for all of their subscribers, or uses a certificate per gateway or region, is wholly up to the carrier.

A carrier of any size is unlikely to issue per-TN certificates, as carriers have hundreds of thousands of numbers leaving and entering their domain per day. Likewise, changing the carrier's certificate, either due to timeout / rotation or unexpected revocation, would require the carrier to regenerate all of their tens or hundreds of millions TNs under management. Even at 2.6ms per TN,[9] a large carrier with a hundred million TNs would need 72 hours to calculate a rollover of their certificates. That does not account for distribution and installation in the authentication servers and publishing the public keys.

Let us come back now to the certificate discussion. This is by no means settled: there were over 40 messages on the STIR list on this in a period of seven days. One manufacturer wants to have a model that supports the SIP UA itself to present the certificate. The reason is they plan on introducing a consumer SIP phone with no serving carrier. As such, a scheme whereby the carrier is the direct vouching entity will not work for them at all. As such, they are vehemently tied to a certificate model that has the number itself encoded and signed in the certificate. From a policy perspective, there would be a chain of trust, rooted at the national numbering authority, to the vendor, to the subscriber.

A per-TN model works great if the private key physically resides in the endpoint or a fixed, local aggregation device. In this situation the private key is in a single box (phone) or a very small number of boxes (DSLAMs, CMTS, inbound SBC farm, etc.). That is not an insurmountable key management problem. However, if the carrier has a model like that depicted in Figure 4, one could imagine that every SBC and CSCF in the carrier's network would need the private keys for all of their numbers. For a large U.S. domestic carrier, that could be north of 100,000,000 keys. Worse yet, standard security practice requires those keys to periodically roll over. If we have a

[9] Informal OpenSSL benchmark on a 2.5GHz quad core i7 generating 2048-bit length public / private key pairs performed at Georgetown.

reasonable key lifetime of three months, that means such a carrier will need to update their keys at a rate of 13/second, presuming no ports or revocations.

Some Other Good Things

There are no obvious holes for sneaking through spoofed CLIDs, unless you have the cooperation of a service provider or the root CA the verifiers happen to use. Presuming the certificate identifies the service provider, the bad service provider gets fingered directly, and the legal system can correct the problem.

There are no obvious holes for typical secure protocol failures. STIR has a replay prevention scheme whereby the user agent is required to have a relatively accurate clock, say to a few seconds or a minute, and STIR includes the Date header field in the signature calculation. This replay prevention scheme is robust against attacks that try to use the signature to forge calls from a number. See below for the hole with respect to calls to a number. STIR also includes the SDP keys of any secured media in the signature, which is a very nice touch. If you care about secure media, then you definitely want to know the media points to the right endpoint and has not been hacked in transit.

Other Open Issues

Policy vs Protocol Action

The proposal documents a number of things as protocol actions that are really policy decisions. Policy, as in what is allowed, whether specified by governments, industry, or users, is something explicitly outside the scope of the IETF. As well, these policy 'requirements' makes a number of the SHOULDs and MUSTs unenforceable. For example, the specified behavior when a verifier receives signed and unsigned calls over the same sending number is a policy issue, not a protocol issue.

Another place where this makes proposal problematic is the specification of the authentication service. By putting it explicitly outside the SIP UA, they have to specify a ton of machinery around how the service authenticates its clients. However, by necessity that has to be underspecified because different providers will have different authentication needs. For example, a DOCSIS operator has to care about neighbors spoofing each other, as the access network is shared media. A DSLAM operator does not have any care at all, because the line identifies the user. A MAN provider may or may not care if there are dedicated facilities or even if the business relationship is strong enough to not need to rely on cryptographic authentication.

This could again be easily fixed by breaking out the implementation from the protocol specification.

Assuming no SIP S/MIME

On the one hand, RFC 3261 solved the problem of authenticating the sender in 2002. If the UAC signs the SIP signaling with their X.509 certificate, we will have a cryptographic assertion of identity (the X.509 subject) and cryptographic integrity verification of the asserted identity in the From field. Moreover, a UAC can sign their SDP, thus providing integrity protection over media ports, keys, and so on.

On the other hand, looking at the participating implementations at the most recent SIPit,[10] there was only a single implementation that offered S/MIME. For all practical purposes, there is no support for S/MIME. As well, issuing certificates to UACs and policies for which root CAs to accept at UASs would still be a problem. The founding principle of STIR is that end users cannot figure out certificate management. As such, service providers will do that for them.

Timeline

We expect STIR to pass through the IETF by the end of 2016. Once the protocol (STIR) has been finalized and implementation guidelines (SHAKEN) completed, implementation will be required in deployable products. Assuming a rollout is a national priority, equipment manufacturers and software developers should have network equipment, phones, and software ready by the end of 2017. If regulators focus the industry to accelerate product development and test, we could see national rollout within two years of product available, which would be by 2020.

Summary

- STIR will help mitigate Caller ID spoofing and the ills enabled by Caller ID spoofing, such as robocalling, SWATting, fraud, and so on.
- STIR is only applicable to calls originated in a SIP network, terminating in a SIP network, and transiting networks that keep the SIP signaling intact. In other words, even if the call originates in SIP and terminates in SIP, if the carriers connect using ISDN PRI or SS7, the STIR signaling will be lost.
- Per-TN certificates, or certificates carrying a range of numbers, provide a direct, cryptographic assertion the signer can sign for a given TN in the From field. This is true so long as the entity issuing root certificates or TN certificates is trusted.
- Certificates over the authentication service (a.k.a. the service provider or colloquially, the “phone company”) provide cryptographic means for regulators to enforce attestations of identity made by the authentication service are accurate.
- A nationwide solution, with focus from service providers, equipment providers, and regulators, could be deployed by 2020.

[10] Sparks, Robert, *SIPit 31 Summary*, https://www.sipit.net/SIPit31_summary September 29 – October 3, 2013, Nice, France