**A****TIS-0x0000x**

ATIS Standard on

**Signature-Based Handling of Asserted Information Using Tokens (SHAKEN):**

**SHAKEN Support of "div" PASSporT Token**

**Alliance for Telecommunications Industry Solutions**

Approved Month DD, YYYY

**Abstract**

The base SHAKEN specification provides replay-detection mechanisms to identify cases where a malicious entity attempts to masquerade as another user by replaying parts of a legitimate INVITE request. However, these mechanisms don’t cover cases where the INVITE is replayed within the short Date freshness window. This technical report describes how the mechanisms defined by draft-ietf-stir-passport-divert can be integrated within the SHAKEN framework to close this replay attack window.

**Foreword**

The Alliance for Telecommunications Industry Solutions (ATIS) serves the public through improved understanding between carriers, customers, and manufacturers. The [**COMMITTEE NAME**] Committee [**INSERT MISSION**]. [**INSERT SCOPE**].

The mandatory requirements are designated by the word *shall* and recommendations by the word *should*. Where both a mandatory requirement and a recommendation are specified for the same criterion, the recommendation represents a goal currently identifiable as having distinct compatibility or performance advantages. The word *may* denotes a optional capability that could augment the standard. The standard is fully functional without the incorporation of this optional capability.

Suggestions for improvement of this document are welcome. They should be sent to the Alliance for Telecommunications Industry Solutions, [**COMMITTEE NAME**], 1200 G Street NW, Suite 500, Washington, DC 20005.

At the time of consensus on this document, [**COMMITTEE NAME**], which was responsible for its development, had the following leadership:

[**LEADERSHIP LIST**]

The **[SUBCOMMITTEE NAME]** Subcommittee was responsible for the development of this document.

**Revision History**

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
|  |  |  |  |

**Table of Contents**

1 Scope & Purpose 1

1.1 Scope 1

1.2 Purpose 1

1.2.1 Document Organization 1

2 Normative References 2

3 Definitions, Acronyms, & Abbreviations 2

3.1 Definitions 2

3.2 Acronyms & Abbreviations 2

4 Overview 3

4.1 SHAKEN Replay-Attack Window 3

4.1.1 Overview of STIR/SHAKEN Replay-Attack Mechanisms 3

4.1.2 Replay Attack within "iat" Freshness Window goes Undetected 3

4.1.3 SIP-PBX Call-Forwarding 4

4.2 Closing the Replay Attack Window 8

4.2.1 Overview of PASSporT "div" Extension 8

4.2.2 "div" PASSporT Token added by Forwarding SP 8

4.2.3 "div" PASSporT Token added by SIP-PBX 10

4.3 Additional Value-add of PASSporT "div" Extension 11

5 SHAKEN support of "div" PASSporT - Normative Requirements 12

A Annex Title 13

**Table of Figures**

Figure 1. Replayed INVITE looks like a legitimately forwarded INVITE 4

Figure 2. SIP-PBX forwards call via new INVITE 6

Figure 3. SHAKEN replay attack window 7

Figure 4. Forwarding Service Provider adds "div" PASSporT token 8

Figure 5. Verifying Chain of Authority across Multiple PASSport Tokens 9

Figure 6. SIP-PBX call forwarding via 302 response containing "div" PASSporT Token 10

Figure 7. SIP-PBX call-forward via new INVITE containing "div" PASSporT Token 11

**Table of Tables**

# Scope & Purpose

## Scope

This document extends SHAKEN to support the PASSporT "div" extension defined in draft-ietf-stir-passport-divert.

## Purpose

The base STIR/SHAKEN replay-attack detection mechanisms are unable to distinguish between a legitimate call that is diverted by a feature such as call-forwarding, and a malicious call where the attacker attempts to masquerade as another user by replaying a legitimate To, From and Identity header within the Date freshness window. This document describes how draft-ietf-stir-passport-divert can be used to close this replay attack window.

### Document Organization

Section 4 provides an informative overview of the replay attack window that exists within the base SHAKEN framework, and describes how the PASSporT "div" extensions can be used to close the window.

Section 5 specifies the normative requirements to add support draft draft-ietf-stir-passport-divert to SHAKEN.

# Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

ATIS-0x0000x, *Technical Report*.

ATIS-0x0000x.201x, *American National Standard*.

# Definitions, Acronyms, & Abbreviations

For a list of common communications terms and definitions, please visit the *ATIS Telecom Glossary*, which is located at < <http://www.atis.org/glossary> >.

## Definitions

**AAA**: xxxx.

**Bbbb**: xxxx.

## Acronyms & Abbreviations

|  |  |
| --- | --- |
| ATIS | Alliance for Telecommunications Industry Solutions |

# Overview

This section provides an overview of the replay attack vulnerabilities that exist within the base SHAKEN framework, and describes how the PASSporT "div" extension defined in draft-ietf-stir-passport-divert can be used to counter these vulnerabilities.

Section 4.1 describes the replay attack window that a malicious entity could use to masquerade as another user

* 4.1.1 provides an overview of the base STIR/SHAKEN replay attack detection mechanisms
* 4.1.2 describes the primary replay-attack window (i.e., replay during "iat" freshness window)
* 4.1.3 describes additional replay attack vulnerabilities associated with SIP-PBXs

Section 4.2 describes how the PASSporT "div" extension can close the replay attack window

* 4.2.1 provides an overview of the PASSporT "div" extension
* 4.2.2 describes how PASSporT "div" closes the replay-attack window
* 4.2.3 describes how PASSporT "div" resolves additional replay attack issues related to SIP-PBXs

## SHAKEN Replay-Attack Window

### Overview of STIR/SHAKEN Replay-Attack Mechanisms

A malicious entity may attempt to masquerade as another user by replaying a valid To, P-Asserted-ID and Identity header in a new INVITE request. STIR defines the procedures that a verification service must apply to detect such an attack. The primary replay-detection mechanism is to verify that the received Date header is within the ~60 second "iat" freshness window. Within this short window, the verification service can perform additional replay checks. First, it can maintain a cache of recently received Identity headers, and verify that it hasn’t received the same Identity header for the same calling number. This works since different valid Identity headers for the same calling TN will always have different "iat" values. Second, if the received INVITE is targeting a voicemail server (i.e., the Request-URI contains the TN used to listen to saved voice messages), then the verification service can verify that the TNs in the Request-URI TN and To header TN match. This works since these calls are never forwarded on their way to the voicemail server, and therefore the Request-URI should always contain the originally dialed TN.

### Replay Attack within "iat" Freshness Window goes Undetected

The SHAKEN verification procedures described above cannot detect all replay attack cases. A replay attack window exists where a malicious entity captures a valid set of To, P-Asserted-Identity and Identity headers from an INVITE to a called user, and replays them within the "iat" freshness window in a new INVITE to different called user. If the Identity header is not stored in the cache of the target service provider, then the base SHAKEN verification procedures have no way to detect that the calling TN has been maliciously spoofed. A verification service cannot detect this case by looking for a mismatch between the called TN in the Request-URI and the originally dialed TN in the To header, since these can mismatch for legitimate call scenarios such as call-forwarding.

Figure 1 illustrates the replay attack window for the case where a malicious entity masquerades as another user by constructing an INVITE request that looks identical to a call-forwarded INVITE.



Figure 1. Replayed INVITE looks like a legitimately forwarded INVITE

Figure 1 shows the initial INVITE message flow for call from TN-a to TN-b that is forwarded to TN-c, where TN-a, TN-b and TN-c are served by SP-a, SP-b, and SP-c respectively. Somewhere along the signaling chain, a malicious entity replays the To, P-Asserted-Identity and Identity headers in a new INVITE to called TN-x. The SHAKEN verification service is unable to distinguish between the legitimate [2] INVITE and the replayed [4] INVITE; they both pass the SHAKEN-defined verification tests. The verification service might be tempted to look for a valid History-Info header that would explain why the To header TN and the Request-URI TN don’t match. But that wouldn’t help since the malicious entity could easily include a similarly valid-looking History-Info header, as shown in [4] INVITE. The root problem is that the INVITE contains no cryptographic proof that the call was legitimately diverted from the original called TN identified in the To header field to a different TN identified in the Request-URI.

Note: While the example in Figure 1 shows the case where authentication information is replayed from an INVITE that is associated with a forwarded call, the attacker could in fact replay valid authentication information from any INVITE (e.g., INVITE associated with 2-way call).

### SIP-PBX Call-Forwarding

When a SIP-PBX forwards a DID call, the calling number should be delivered to the forward-to service provider network, along with the Identity header added by the SHAKEN authentication service in the originating network. For example, when a call is forwarded from my office number to my mobile number, my mobile phone should display both the actual calling number, and an accurate indication of the legitimacy of the calling number based on the SHAKEN verification results. In order to make this work, the SIP-PBX must behave in a predictable way. The current industry standard for SIP interworking between the SIP-PBX and its host service provider is SIPconnect 2.0.

#### SIPconnect 2.0 Call-Forwarding Procedures

SIPconnect 2.0 defines two call-forwarding procedures for DID calls; the SIP-PBX can forward the call either by responding to the incoming INVITE request with a 302 Moved Temporarily response that redirects the call to the forward-to number, or by sending a new INVITE request to the forward-to number.

For the 302 response case, the host service provider consumes the response, and forwards the call by updating the INVITE Request-URI to identify the forward-to user. This creates an INVITE that looks similar to the forwarded [2] INVITE shown previously in Figure 1; i.e., an INVITE that is indistinguishable from a replayed INVITE.

For the INVITE case, the SIP-PBX must populate the new INVITE request as follows:

* The Request-URI must contain the forward-to TN
* The From header must contain the original calling TN
* The P-Asserted-Identity header must contains the SIP-PBX forwarding TN
* The History-Info header must record the forwarding event

SIPconnect 2.0 does not place any requirements on populating the To header, so presumably it contains the original called TN.

In order to convey the SHAKEN PASSporT token end-to-end, a SIP-PBX that forwards calls using the "new INVITE" mechanism must relay the received Identity header back to the host service provider in the forwarding INVITE request. If the SIP-PBX does convey the Identity header intact, there are still two issues that need to be resolved:

1. If the SIP-PBX updates the P-Asserted-Identity header as mandated by SIPconnect 2.0, it will break the SHAKEN PASSporT token signature, since SHAKEN verification uses the contents of P-Asserted-Identity to create the local "orig" claim during signature validation.
2. The above problem can be avoided by having the host service provider update the P-Asserted-Identity header sent to the forward-to network so that it contains the original calling TN. However, this has the disadvantage that it provides the SIP-PBX with a replay attack entry point into the network.

#### SIP-PBX Call-Forwarding Example

These issues are illustrated in Figure 2.



Figure 2. SIP-PBX forwards call via new INVITE

Figure 2 shows the INVITE sequence for the call scenario where TN-a served by SP-a calls TN-b served by a SIP-PBX hosted by SP-b, and TN-b forwards the DID call to TN-c hosted by SP-c. The message sequence is as follows:

1. SP-a sends [1] INVITE to SP-b to establish a call from TN-a to TN-b. The INVITE contains an Identity header that digitally signs calling TN-a. On receiving [1] INVITE, SP-b verifies the Identity header.
2. SP-b routes the received INVITE, including the verified Identity header, to PBX-1. Since called TN-b has activated call-forwarding to TN-c, PBX-1 updates the INVITE Request-URI to identify forward-to user TN-c, and (per SIPconnect 2.0) the P-Asserted-Identity header to identify the forwarding user TN-b.
3. PBX-1 sends the updated INVITE, including the received Identity header, back to SP-b. There is no explicit indication in the [3] INVITE that it is establishing the forwarded leg of a call. Therefore, SP-b can handle this INVITE using one of the following two options:
4. SP-b assumes that the received [3] INVITE is a normal originating call from PBX-1. It verifies that the calling TN in the P-Asserted-Identity header (TN-b) belongs to the set of TNs delegated to the PBX-1. It then performs SHAKEN authentication for calling TN-b. Since the To header also contains TN-b, this creates the odd situation where the "orig" and "dest" claims of the resulting PASSporT token contain the same telephone number. SP-b adds a second Identity header to the INVITE containing the signed output of the authentication service, and sends [4a] INVITE to SP-c.

On receiving [4a] INVITE, SP-c verifies the two Identity headers. Verification fails for the Identity header associated with calling TN-a. Verification passes for the second Identity header associated with TN-b (although SP-c may consider it a "fail" because the "orig" and "dest" TNs match). The best SP-c can do is deliver the forwarding TN-b as the authenticated calling TN to the called user (which is incorrect since TN-a is the calling TN, not TN-b).

1. Based on the presence and contents of various headers (e.g., Identity, From, History-Info) SP-b assumes that [3] INVITE is establishing the forward-to leg of a forwarded call. Knowing that SIPconnect 2.0 compliant PBX-1 has broken the SHAKEN Identity PASSporT signature, SP-b fixes the signature by updating the P-Asserted-Identity header to contain the same TN as the PASSporT "orig" claim, and sends [4b] INVITE to SP-c.

On receiving [4b] INVITE, SP-c verifies the Identity header. Verification passes, and SP-c delivers the correct calling TN-a to the called user.

#### SIP-PBX Replay Attack Entry Point

Of the two call-forwarding options described in the previous section, option-2 is obviously the better choice, since it delivers the correct calling TN to the forward-to user. However, the SIP-PBX could easily take advantage of option-2 to maliciously masquerade as another user, as shown in Figure 3. Therefore, option-2 would only apply for cases where the host service provider has a high level of trust with the SIP-PBX. This leaves a gap in terms of how to detect replay attacks from a SIP-PBX for cases where the requisite level of trust does not exist.



Figure 3. SHAKEN replay attack window

In Figure 3, PBX-1 attempts to masquerade as TN-a by replaying the To, P-Asserted-Identity and Identity headers from received [1] INVITE into request [6] INVITE sent to host SP-b. Following option-2 described above, SP-b "fixes" the P-Asserted-Identity header, so that when [7] INVITE arrives at SP-x, verification passes and the malicious call goes undetected.

## Closing the Replay Attack Window

This section describes how the PASSporT "div" extension defined in ietf-draft-stir-passport-divert can be used to close the SHAKEN replay attack window.

### Overview of PASSporT "div" Extension

ietf-draft-stir-passport-divert defines a PASSporT "div" extension ("ppt":"div") that enables the STIR authentication service of the forwarding service provider to cryptographically sign the forwarding TN. The "div" PASSporT token contains the following four claims:

* "orig" – contains the calling number identified by the base SHAKEN "orig" claim
* "dest" – the destination number after retargeting
* "iat" – the current date/timestamp
* "div" – the destination number before retargeting

The forwarding service provider must be authoritative for the TN contained in the "div" claim. The forwarding SP generates a signature for the token using the private key of its STI certificate, and includes an "x5u” parameter in the token that references the STI-CR file containing the STI certificate. Verifiers can then obtain the certificate to verify the "div" PASSporT signature, and thus providing cryptographic proof that the call was legitimately diverted.

### "div" PASSporT Token added by Forwarding SP

Figure 4 shows how the divert PASSporT extension closes the replay attack window described earlier in Figure 1.



Figure 4. Forwarding Service Provider adds "div" PASSporT token

As in Figure 1, Figure 4 shows the initial INVITE message sequence for a call from TN-a to TN-b that is forwarded to TN-c. Before forwarding the call, SP-b adds a PASSporT "div" token to [2] INVITE to provide cryptographic proof that the call is being legitimately forwarded from TN-b to TN-c. Meanwhile, a malicious entity attempts to masquerade as TN-a by replaying the To, P-Asserted-Identity, Date and Identity headers from [2] INVITE into a new [4] INVITE to TN-x. The SHAKEN verification services in SP-c and SP-x can distinguish between the legitimate call-forwarded call and the malicious call by verifying that the "shaken" and "div" PASSporT tokens provide an unbroken chain of authority between the final called TN identified in the Request-URI and the initial dialed TN identified in the "dest" claim of the "shaken" PASSporT token. In this example, the SHAKEN verification service in SP-x detects that the chain in [4] INVITE is broken, since the Request-URI TN does not match the “dest” claim in the “div” PASSporT token. As a result, SP-x includes a “fraud alert” indication in the [5] INVITE request to called UE-x.

An INVITE that is forwarded multiple times would have multiple "div" PASSporT tokens; one for each forwarding event. The verification service that receives such an INVITE will arrange the "div" PASSporT tokens in order, and verify the chain of authority from the Request-URI TN, through the multiple "div" PASSporT tokens to the "dest" TN in the "shaken" PASSporT token. Figure 5 illustrates the verification process for the case where TN-a calls TN-b, and the call is forward twice; first to TN-c, and then to TN-d.



Figure 5. Verifying Chain of Authority across Multiple PASSport Tokens

### "div" PASSporT Token added by SIP-PBX

A SIP-PBX can use a "div" PASSporT token to close the replay attack entry point over a SIP Trunk interface. When forwarding a call, the PBX generates the “div” PASSporT token using its PoP certificate, and includes it in an Identity header in the 302 response or new INVITE sent back to the host service provider. The service provider uses the received "div" PASSporT to verify the chain of authority from the forward-to-TN to the originally called TN. Once verified, the host service provider replaces the PBX’s "div" PASSporT with a "div" PASSporT that has the same set of claims, but is signed using the host service provider’s STI certificate.

Figure 6 shows how the "div" PASSporT token is used when a PBX call is forwarded using the 302 Moved Temporarily response.



Figure 6. SIP-PBX call forwarding via 302 response containing "div" PASSporT Token

Figure 7 shows how the "div" PASSporT token is used when a PBX call is forwarded using a new INVITE request.



Figure 7. SIP-PBX call-forward via new INVITE containing "div" PASSporT Token

## Additional Value-add of PASSporT "div" Extension

The base SHAKEN verification procedure uses the To header TN to build the local "dest" claim that forms part of the input to the signature validation function. Therefore, if a call diversion feature updates the To header TN after the INVITE leaves the originating network (i.e., after SHAKEN authentication has occurred), it breaks the SHAKEN PASSporT signature, resulting in a false-negative verification failure. (Updating To header is rare, although there is at least one feature in the MMTEL spec that does this).

The PASSporT "div" extension resolves this issue, since the divert PASSporT token added during call diversion will provide verification services with sufficient information to accurately create the local "dest" claim for the SHAKEN PASSporT token.

# SHAKEN support of "div" PASSporT - Normative Requirements

***<… under construction …>***

This section defines the normative requirements to add support of ietf-draft-stir-passport-divert to the SHAKEN framework.

*Open issues for discussion:*

1. *Issue: For SHAKEN, does the "div" PASSporT token contain an "origid" claim? If yes, we would need to define yet another PASSporT extension to add the "origid" claim to the existing set of "div" claims.*

*Recommendation: Don’t extend "div" PASSporT to include "origid" claim, since cost would be high (need to write yet another IETF draft) while benefit would be minimal.*

1. *Issue: The divert draft provides two ways to convey the multiple PASSporT tokens:*

* *Use multiple Identity headers, one token per header, or…*
* *Nest the multiple PASSporT tokens in a single Identity header; i.e., nest the SHAKEN PASSporT token inside the first "div" PASSporT token, nest all of that inside the 2nd "div" PASSporT token, etc.*

*The 2nd option has the advantage that the nesting arrangement explicitly indicates the order of the divert PASSporT tokens, so makes it easier for verification services to create the chain of authority. However, if the divert extension is rolled out gradually across service providers, there would be an issue where verification services of providers that haven’t implemented the extension yet wouldn’t be able to see the base SHAKEN PASSport token nested inside the divert PASSPorT token. So, we would need to manage this somehow; e.g., a) don’t support the nested option (at least until everyone supports it), or b) each SP that supports divert PASSporT would choose the option that is supported by each peer SP.*

*Recommendation: Don’t support the nesting option (at least initially) for SHAKEN. Reason being – even if an authentication service carefully only does nesting for peering partners that support nesting, there’s always the chance that an INVITE with a nested Identity header is forwarded to a provider that doesn’t support nesting.*

1. *Issue: …*

# A Annex Title

Xxx