SecSIPIdX
Library, CLI tool and RESTApi server for STIR/SHAKEN

Federico Cabiddu
Voice Lead Developer @Libon
Kamailio Project Contributor
federico.cabiddu@gmail.com

Daniel-Constantin Mierla
Co-Founder Kamailio Project
daniel@asipto.com
@miconda

STIR - SHAKEN

STIR (Secure Telephony Identity Revisited)

- a series of IETF RFCs: RFC8224, 8225, 8226
- https://tools.ietf.org/html/rfc8224

SHAKEN (Secure Handling of Asserted information using toKENs)

- RFC8588 - https://tools.ietf.org/html/rfc8588

They defines how telephone service providers should work together to ensure calling numbers have not been spoofed.

The name was inspired by Ian Fleming's character James Bond, who famously prefers his martinis "shaken, not stirred." STIR having existed already, the creators of SHAKEN "tortured the English language until [they] came up with an acronym."

https://en.wikipedia.org/wiki/STIR/SHAKEN

How does STIR/SHAKEN work?

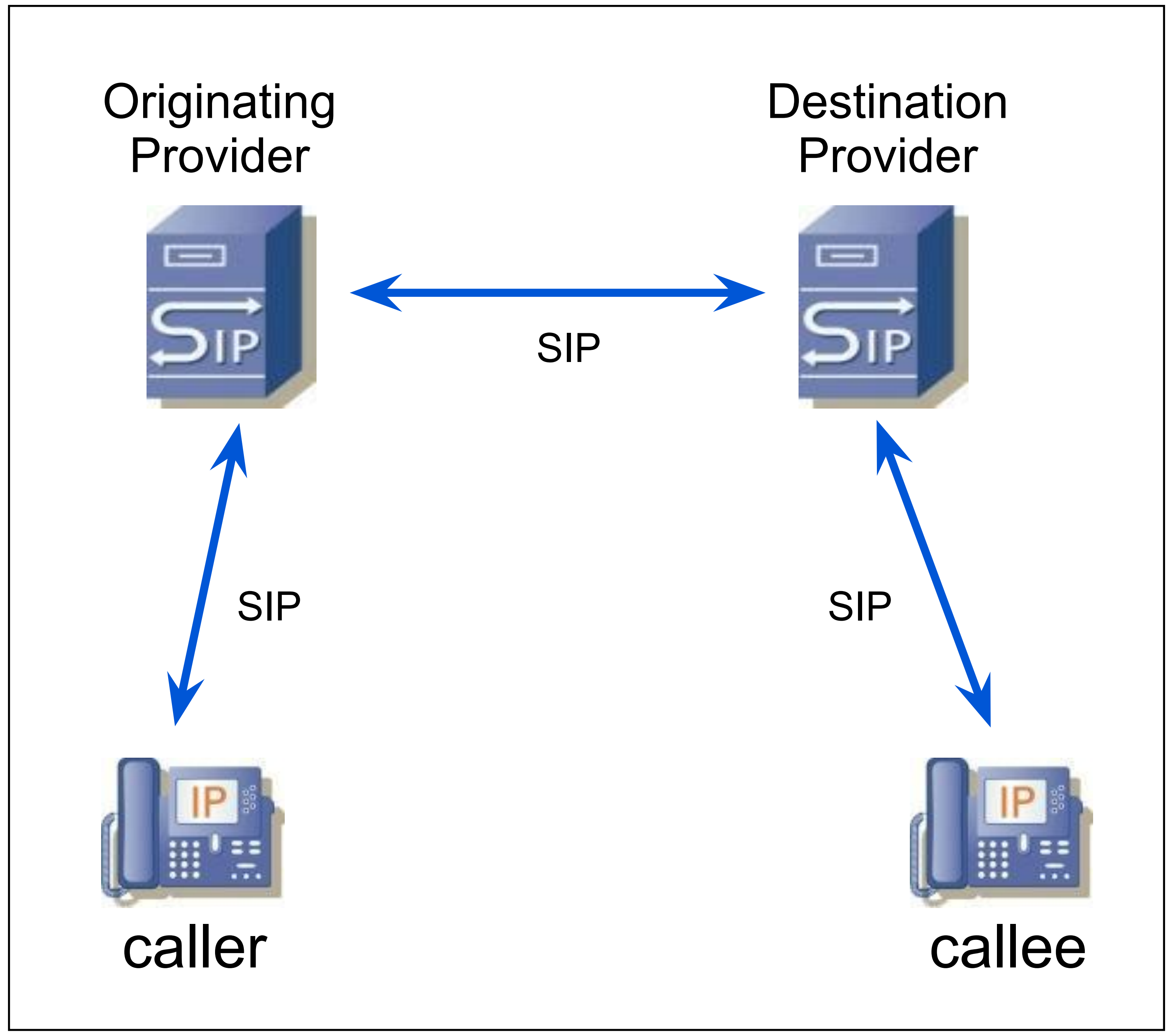
1. When a call is initiated, a SIP INVITE is received by the originating service provider.

2. The originating service provider verifies the call source and number to determine how to confirm validity.
    - ***Full Attestation (A)*** — The service provider authenticates the calling party AND confirms they are authorized to use this number. An example would be a registered subscriber.
    - ***Partial Attestation (B)*** — The service provider verifies the call origination but cannot confirm that the call source is authorized to use the calling number. An example would be a calling number from behind an enterprise PBX.
    - ***Gateway Attestation (C)*** — The service provider authenticates the call's origin but cannot verify the source. An example would be a call received from an international gateway.

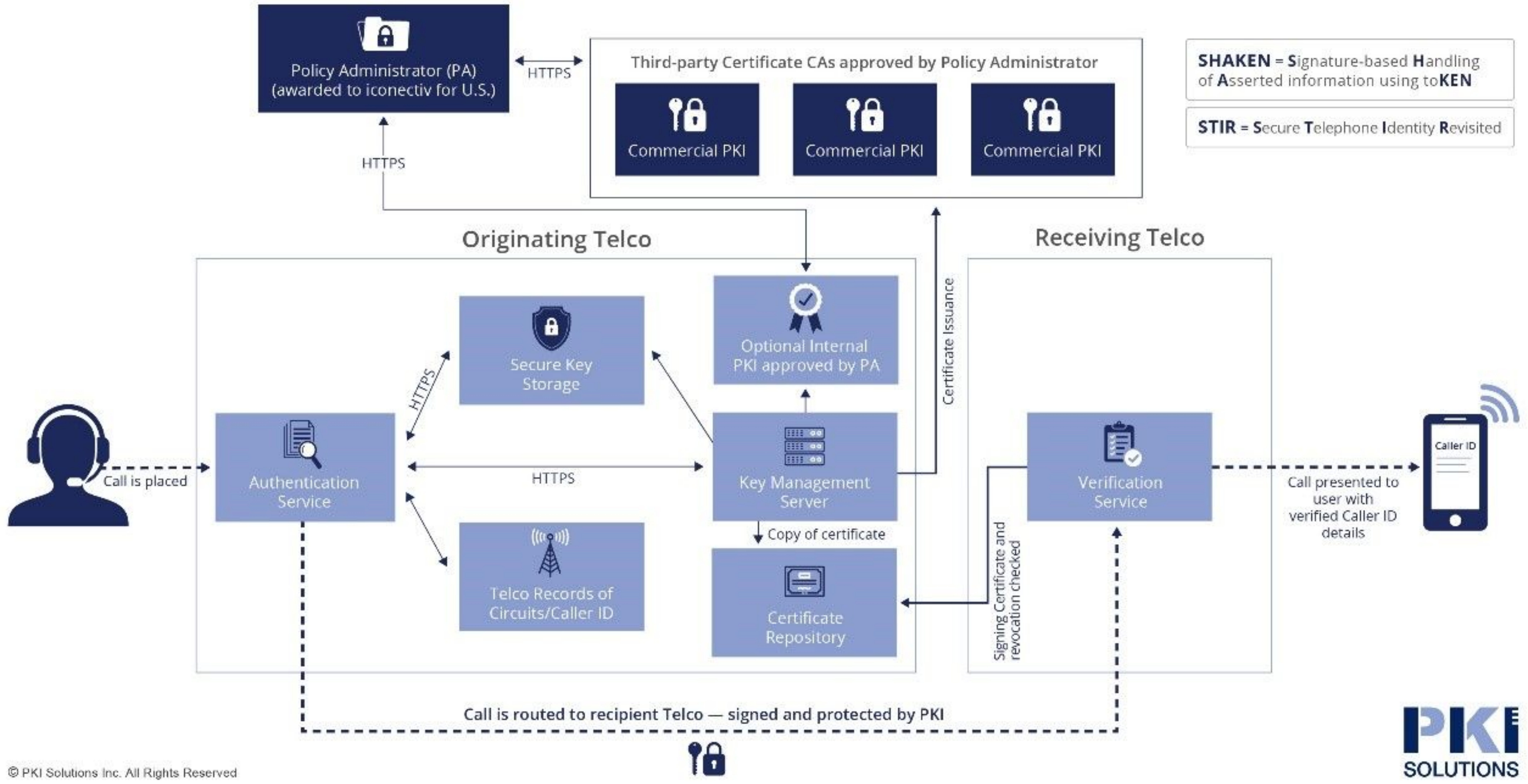
3. The originating service provider will now create a **SIP Identity** header that contains information on the calling number, called number, attestation level, and call origination, along with the certificate.

4. The SIP INVITE with the **SIP Identity** header with the certificate is sent to the destination service provider.

5. The destination service provider verifies the identity of the header and certificate.



Global SHAKEN/STIR Framework



Policy Administrator (PA) (awarded to iconectiv for U.S.)
HTTPS
Third-party Certificate CAs approved by Policy Administrator
Commercial PKI
Commercial PKI
Commercial PKI
HTTPS
Originating Telco
Receiving Telco
Certificate Issuance
Secure Key Storage
Optional Internal PKI approved by PA
HTTPS
Call is placed
Authentication Service
HTTPS
Key Management Server
Verification Service
Caller ID
Call presented to user with verified Caller ID details
Copy of certificate
Signing Certificate and revocation checked
Telco Records of Circuits/Caller ID
Certificate Repository
Call is routed to recipient Telco — signed and protected by PKI

SHAKEN = **S**ignature-based **H**andling of **A**sserted information using to**KEN**

STIR = **S**ecure **T**elephone **I**dentity **R**evisited

PKI SOLUTIONS

Where is it used?

- US
- Canada
- France (“MAN”)

SIP Identity Header

```
INVITE sip:+493055559999@asipto.lab SIP/2.0
Via: SIP/2.0/UDP 192.168.178.75:65337;branch=z9hG4bK.39835377;rport;alias
From: <sip:+493044448888@asipto.lab>;tag=7076833e
To: <sip:+493055559999@asipto.lab>
Call-ID: 1886815038@192.168.178.75
CSeq: 1 INVITE
Contact: <sip:192.168.178.75:65337>
Content-Length: 800
Max-Forwards: 70
Identity:
eyJhbGciOiJFUzI1NiIsInBwdCI6InNoYWtlbiIsInR5cCI6InBhc3Nwb3J0IiwieDV1IjoiaHR0cDovL2FzaXB0by5sYWIvc3
Rpci9jZXJ0LnBlbSJ9.eyJhdHRlc3QiOiJBIiwiZGVzdCI6eyJ0biI6WyI0OTMwNTU1NTk5OTkiXX0sImlhdCI6MTU4MD
ExNTE1Nywib3JpZyI6eyJ0biI6IjQ5MzA0NDQ0ODg4OCJ9LCJvcmlnaWQiOiJhZWRmNmI2MS1kMWM3LTQ3YTEt
ODhjNi03NTBlOThmNDdiZTUifQ.Mwd84gpf_IE0C3VQ1rLtQHyVEBkcmzNXyOqOkrx118nTsGpPKVdLDhFjoJVx4Z
DQ-UZ0ey_Rjw8K2I2hz2kJkw;info=<http://asipto.lab/stir/cert.pem>;alg=ES256;ppt=shaken
…
```

SIP Identity Header

Identity:

eyJhbGciOiJFUzI1NiIsInBwdCI6InNoYWtlbiIsInR5cCI6InBhc3Nwb3J0IiwieDV1IjoiaHR0cDovL2FzaXB0b
y5sYWIvc3Rpci9jZXJ0LnBlbSJ9.eyJhdHRlc3QiOiJBIiwiZGVzdCI6eyJ0biI6WyI0OTMwNTU1NTk5OTkiXX
0sImlhdCI6MTU4MDExNTE1Nywib3JpZyI6eyJ0biI6IjQ5MzA0NDQ0ODg4OCJ9LCJvcmlnaWQiOiJhZWR
mNmI2MS1kMWM3LTQ3YTEtODhjNi03NTBlOThmNDdiZTUifQ.Mwd84gpf_IE0C3VQ1rLtQHyVEBkcmz
NXyOqOkrx118nTsGpPKVdLDhFjoJVx4ZDQ-UZ0ey_Rjw8K2I2hz2kJkw;info=<http://asipto.lab/stir/cert.
pem>;alg=ES256;ppt=shaken

The SIP Identity header contains:

- a JSON web token (JWT - https://en.wikipedia.org/wiki/JSON_Web_Token)
- three parameters

The JWT has three sections:

- header
- payload
- signature.

The header and payload are Base64-URL encoded JSON.

JWT - Header Part

Base64-URL

eyJhbGciOiJFUzI1NiIsInBwdCI6InNoYWtlbiIsInR5cCI6InBhc3Nwb3J0Iiwi
eDV1IjoiaHR0cDovL2FzaXB0by5sYWIvc3Rpci9jZXJ0LnBlbSJ9

decoded

```
{
    "alg": "ES256",
    "ppt": "shaken",
    "typ": "passport",
    "x5u”: "http://asipto.lab/stir/cert.pem"
}
```

The json attributes:

- **alg** — the *encryption algorithm* - **ES256**.
- **ppt** — the *extension* used - **shaken**.
- **typ** — the *token type* - **passport**.
- **x5u** — the *location of the certificate* used to sign the token.

JWT - Payload Part

Base64-URL

eyJhdHRlc3QiOiJBIiwiZGVzdCI6eyJ0biI6WyI0OTMwNTU1NTk5OTkiXX0sImlhdCI6MTU4MDExNTE1Nywib3J
pZyI6eyJ0biI6IjQ5MzA0NDQ0ODg4OCJ9LCJvcmlnaWQiOiJhZWRmNmI2MS1kMWM3LTQ3YTEtODhjNi03N
TBlOThmNDdiZTUifQ

decoded

```
{
    “attest":"A",
    “dest”: {
        “tn": [ “493055559999" ]
     },
     “iat":1580115157,
     “orig":{"tn":"493044448888"},
     “origid":"aedf6b61-d1c7-47a1-88c6-750e98f47be5"
}
```

The json attributes:

- **attest** — the attestation level. Must be either **A**, **B**, or **C**.
- **dest** — the called number(s) or called Uniform Resource Identifier(s).
- **iat** — the timestamp when the token was created.
- **orig** — the calling number or calling Uniform Resource Identifier.
- **origid** — the origination identifier.

Attestation Levels

There are three levels of verification, or “attestation”:

- **A -** the highest level - "Full Attestation” - indicates that the provider recognizes the entire phone number as being registered with the originating subscriber. This would be the case for a landline or mobile phone where the customer connects directly to the VOIP network and the phone number can be verified as being a particular customer, or in the case of a company that has registered a particular callback number.
- **B** - “Partial Attestation” - indicates that the call originated with a known customer but the entire number cannot be verified, which would be the case with a call originating from a client PBX where the extension number is not registered with the provider
- **C** - “Gateway Attestation” - indicates the call can only be verified as coming from a known gateway, for instance, a connection to another service provider.

JWT - Signature Part

Base64-URL ⇨

Mwd84gpf_IE0C3VQ1rLtQHyVEBkcmzNXyOqOkrx118nTsGpP

KVdLDhFjoJVx4ZDQ-UZ0ey_Rjw8K2I2hz2kJkw

Base64URL(ES256 (Base64URL(JWTHeader).Base64URL(JWTPayload)))

SIP Identity Header - Parameters

```
info=<http://asipto.lab/stir/cert.pem>;alg=ES256;ppt=shaken
```

The SIP Identity parameters:

- **info** - the location of the certificate used to sign the token - must be the same as the x5u value in the JWT
- alg - the encryption algorithm to build the signature - must be **ES256**.
- ppt - the extension used - must be **shaken**

STIR/SHAKEN Libraries

Two opensource projects

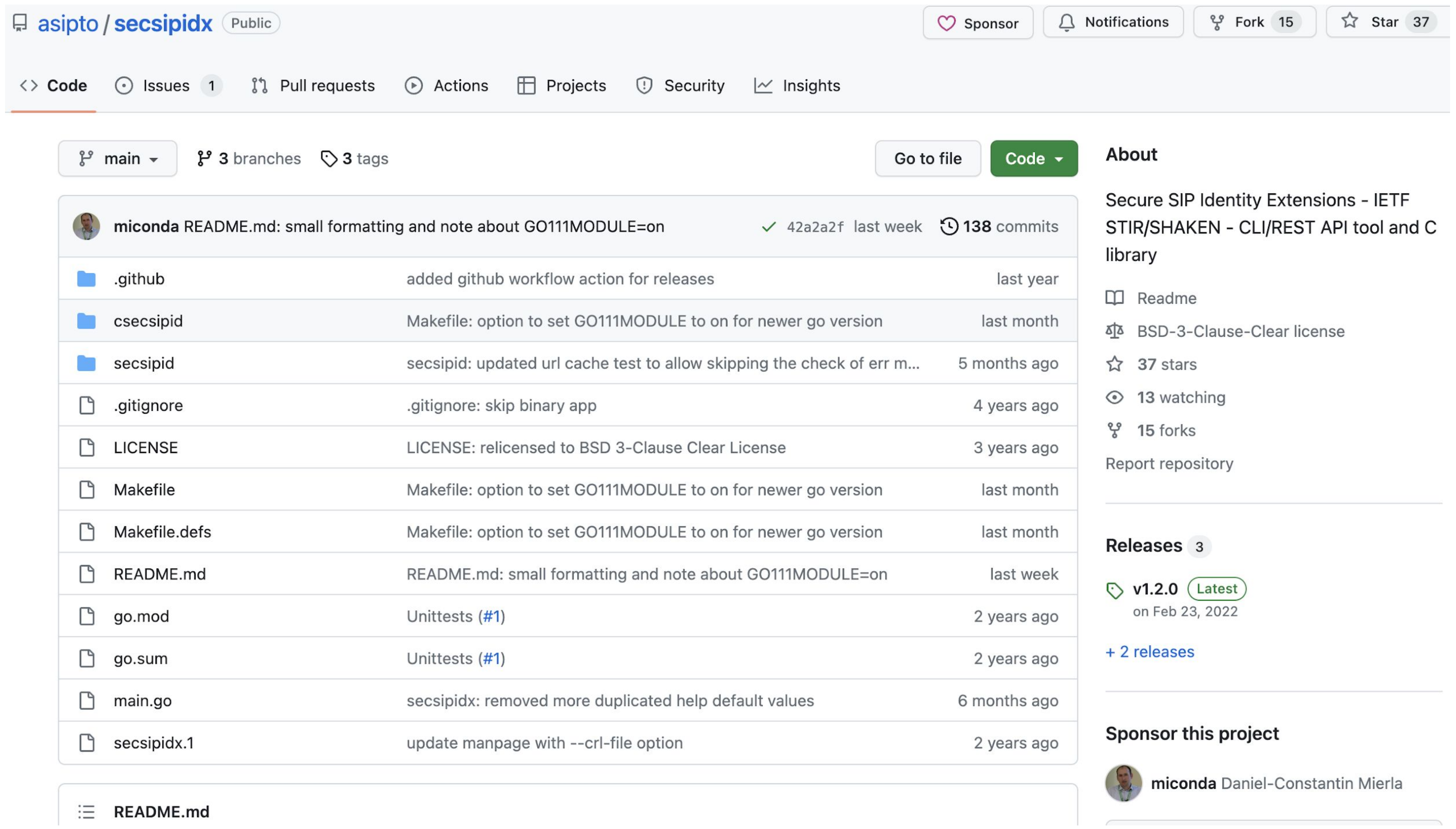
- SignalWire libstirshaken: https://github.com/signalwire/libstirshaken
- SecSIPIDx: https://github.com/asipto/secsipidx

SecSIPIDx Project

https://github.com/asipto/secsipidx

Components:

- secsipid: Go library - common functions
- csecsipid: C library - wrapper code to build dynamic or static library and .h include files
- secsipidx: main.go - CLI tool and HTTP API server for checking or building SIP identity



asipto / secsipidx Public

Sponsor · Notifications · Fork 15 · Star 37

Code · Issues 1 · Pull requests · Actions · Projects · Security · Insights

main · 3 branches · 3 tags · Go to file · Code

miconda README.md: small formatting and note about GO111MODULE=on ✓ 42a2a2f last week 138 commits

.github	added github workflow action for releases	last year
csecsipid	Makefile: option to set GO111MODULE to on for newer go version	last month
secsipid	secsipid: updated url cache test to allow skipping the check of err m...	5 months ago
.gitignore	.gitignore: skip binary app	4 years ago
LICENSE	LICENSE: relicensed to BSD 3-Clause Clear License	3 years ago
Makefile	Makefile: option to set GO111MODULE to on for newer go version	last month
Makefile.defs	Makefile: option to set GO111MODULE to on for newer go version	last month
README.md	README.md: small formatting and note about GO111MODULE=on	last week
go.mod	Unittests (#1)	2 years ago
go.sum	Unittests (#1)	2 years ago
main.go	secsipidx: removed more duplicated help default values	6 months ago
secsipidx.1	update manpage with --crl-file option	2 years ago

README.md

About

Secure SIP Identity Extensions - IETF STIR/SHAKEN - CLI/REST API tool and C library

Readme

BSD-3-Clause-Clear license

37 stars

13 watching

15 forks

Report repository

Releases 3

v1.2.0 Latest on Feb 23, 2022

+ 2 releases

Sponsor this project

miconda Daniel-Constantin Mierla

SecSIPIDx - Standalone Project & Golang

- Playing with the idea of developing extensions for Kamailio in Go language
- HTTP API service to use many SIP server nodes
- Embedded HTTP/S client (not only HTTP/S server)
- Command line tool - could be handy for troubleshooting (or exec() from old fashioned apps)
- Easier to reuse by others and contribute, not tight to Kamailio project
- An easier way to integrate with old deployments (e.g., older version of Kamailio)

Secsipidx

CLI - Generate Full Identity Header

A call from +493044448888 to +493055559999 with attestation level A, when the public key can be downloaded from http://asipto.lab/stir/cert.pem:

```
secsipidx -sign-full -orig-tn 493044448888 -dest-tn 493055559999 -attest A -x5u http://asipto.lab/stir/cert.pem -k ec256-private.pem
```

CLI - Check Full Identity Header

Check the identity header stored in file identity.txt using the public key in file ec256-public.pem with token expire of 3600 seconds:

```
secsipidx -check -fidentity identity.txt -fpubkey ec256-public.pem -expire 3600
```

HTTP Server

Run secsipidx as an HTTP server listening on port 8090 for checking SIP identity with public key from file ec256-public.pem:

```
secsipidx -http-srv ":8090" -http-dir /secsipidx/http/public -fprvkey ec256-private.pem -fpubkey ec256-public.pem -expire 3600
    -timeout 5
```

To run secsipidx as an HTTPS server on port 8093, following command line parameters have to be provided:

```
secsipidx -https-srv ":8093" -https-pubkey /keys/secsipidx-public.key  -https-prvkey /keys/secsipidx-private.key ...
```

Secsipidx HTTP API

Check Identity

If the identity header body is saved in the file identity.txt, the next command can be used to check it:
curl --data @identity.txt http://127.0.0.1:8090/v1/check

If secsipidx is started without -fpubkey or -pubkey, then the public key to check the signature is downloaded from x5u URL (or the header info parameter). The value of -timeout parameter is used to limit the download time of the public key via HTTP.

Generate Identity - CSV API

Prototype:
curl --data 'OrigTN,DestTN,ATTEST,OrigID,X5U' http://127.0.0.1:8090/v1/sign-csv

If OrigID is missing, then a UUID value is generated internally.

Example to get the Identity header value:
curl --data '493044442222,493088886666,A,,https://asipto.lab/v1/pub/cert.pem' http://127.0.0.1:8090/v1/sign-csv

HTTP File Server

When started with parameter -httpdir, the secsipidx servers the files from the respective directory on the URL path /v1/pub/

Kamailio - SecSIPID Module

https://www.kamailio.org/docs/modules/devel/modules/secsipid.html

- `secsipid_check_identity(keyPath)`
- `secsipid_check_identity_pubkey(pubkeyVal)`
- `secsipid_check(sIdentity, keyPath)`
- `secsipid_get_url(url, ovar)`
- `secsipid_add_identity(origTN, destTN, attest, origID, x5u, keyPath)`
- `secsipid_build_identity(origTN, destTN, attest, origID, x5u, keyPath)`
- `secsipid_build_identity_prvkey(origTN, destTN, attest, origID, x5u, keyData)`
- `secsipid_sign(sheaders, spayload, keyPath)`
- `secsipid_sign_prvkey(sheaders, spayload, keyData)`

Kamailio - SecSIPID Module

```
loadmodule “secsipid.so”
...
modparam("secsipid", "expire", 600)
modparam("secsipid", "timeout", 5)
...
request_route {
    ...
    if(secsipid_check_identity("/secsipid/$si/cert.pem")) { ... }
    ...
    if(secsipid_check_identity("")) { ... }
    ...
    secsipid_add_identity("$fU", "$rU", "A", “",
            “http://kamailio.org/stir/$rd/cert.pem",
"/secsipid/$rd/key.pem"));
    ...
}
...
```

THANK YOU!

KAMAILIO WORLD
CONFERENCE & EXHIBITION

April 18-19, 2024
Berlin, Germany
https://www.kamailioworld.com/k2024/