DP3TAPI

Documentation

April 20, 2020

Contents

Technical Description	2
Introduction	2
Verification of Data	2
Google/Apple Privacy-Preserving Contact Tracing Similarities	2
Web Service	3
Introduction	3
/v 1 /	3
Responses	3
/v1/exposed	3
Request Headers	3
Responses	3
200 Success	3 4
/v1/exposed/dayDateStr	4
Path Parameters	4
Responses	4
200 Success	4 4
	•
Models	4
ExposedOverview	4
Exposee	
ExposeeAuthData	
ExposeeRequest	5

Technical Description

Introduction

This document outlines the backend as it is. The models and requests are automatically generated. Hence, they should reflect the current live situation. We try to provide examples and description to clarify the use of the fields and responses returned.

Verification of Data

To handle heavy workload, requests are routed via a content-delivery-network (CDN). This means that we need to provide proof that the data was not modified by the CDN. We propose a Elliptic-Curve Digital Signature Algorithm using the P256 elliptic curve with a SHA-256 hashing algorithm. The P256 elliptic curve has good native support for the Apple and Android platforms to verify signatures. The public key should be available on the discovery platform and is as well included and distributed with the applications for iOS and Android.

The ensure the possibility of signature verification, the signed endpoints return an object with a signature and a data field, of which the data field contains a base64 representation of the list. In the current implementation the representation is a json of a list of keys. To improve performance of possible large decodings, we plan to switch to protobuf or something similar, which should speed up the parsing.

Since we only want to ensure that the data we are processing was indeed the data sent from the specified backend, it is sufficient to generate the signature of the content which will be processed.

Too further improve operability, the algorithm used to generate the signature should as well be encoded within the json object, similiar to a JWK (Json web key).

Google/Apple Privacy-Preserving Contact Tracing Similarities

Web Service

Introduction

A test implementation is hosted on: https://demo.dpppt.org.

This part of the documentation deals with the different API-Endpoints. Examples to fields are put into the models section to increase readability. Every request lists all possible status codes and the reason for the status code.

/v1/

get /v1/

hello

Responses

200 Success

Туре	
string	

/v1/exposed

post /v1/exposed

Enpoint used to publish the SecretKey.

Request Headers

Field	Туре	Description
User-Agent *	string	App Identifier (PackageName/BundleIdentifier) + App-Version + OS (Android/iOS) + OS-Version

Request Body

Field	Туре	Description
*	ExposeeRequest	The ExposeeRequest contains the SecretKey from the guessed infection date, the infection date itself, and some authentication data to verify the test result

Responses

200 Success

Returns OK if successful

Туре	
string	

400 Bad Request

Key is not base64 encoded

/v1/exposed/dayDateStr

get /v1/exposed/{dayDateStr}

Path Parameters

Field	Туре	Description
dayDateStr * string		The date for which we want to get the SecretKey.

Responses

200 Success

Returns ExposedOverview, which includes all secretkeys which were published on dayDateStr.

Туре	
ExposedOverview	

400 Bad Request

If dayDateStr has the wrong format

Models

All Models, which are used by the Enpoints are described here. For every field we give examples, to give an overview of what the backend expects.

ExposedOverview

Field	Type	Description	Example
exposed	Exposee[]	A list of all SecretKeys	c.f.Exposeemodel

Exposee

Field	Туре	Description	Example
key *	string	The SecretKey of a exposed as a base64 encoded string. The SecretKey consists of 32 bytes.	QUJDREVGR0hJS ktMTU5PUFFSU1 RVVIdYWVpBQkN ERUY=
onset *	string	The onset of an exposed.	2020-04-06

${\sf Exposee Auth Data}$

Field	Туре	Description	Example
value	string	Authentication data used to verify the test result (base64 encoded)	TBD

ExposeeRequest

Field	Туре	Description	Example
key *	string	The SecretKey used to generate EphID base64 encoded.	QUJDREVGR0hJS ktMTU5PUFFSU1 RVVIdYWVpBQkN ERUY=
onset *	string	The onset date of the secret key. Format: yyyy-MM-dd	2019-01-31
authData *	ExposeeAuthData	AuthenticationData provided by the health institutes to verify the test results	TBD