FIDO Metadata Service

Proposed Standard, May 18, 2021

This version:
http://fidoalliance.org/specs/mds/fido-metadata-service-v3.0-ps-20210518.html

Issue Tracking:
GitHub

Editors:
Billy Jack (Microsoft)
Rolf Lindemann (Nok Nok Labs)
Yuriy Ackermann (FIDO Alliance)

Abstract

The FIDO Authenticator Metadata Specification defines so-called "Authenticator Metadata" statements. The metadata statements contains the "Trust Anchor" required to validate the attestation object, and they also describe several other important characteristics of the authenticator. The metadata service described in this document defines a baseline method for relying parties to access the latest metadata statements.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current FIDO Alliance publications and the latest revision of this technical report can be found in the FIDO Alliance specifications index at https://www.fidoalliance.org/specifications/.

This document was published by the FIDO Alliance as a Proposed Standard. If you wish to make comments regarding this document, please Contact Us. All comments are welcome.

Implementation of certain elements of this Specification may require licenses under third party intellectual property rights, including without limitation, patent rights. The FIDO Alliance, Inc. and its Members and any other contributors to the Specification are not, and shall not be held, responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

THIS FIDO ALLIANCE SPECIFICATION IS PROVIDED “AS IS” AND WITHOUT ANY WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY EXPRESS OR IMPLIED WARRANTY OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

This document has been reviewed by FIDO Aliance Members and is endorsed as a Proposed Standard. It is a stable document and may be used as reference material or cited from another document. FIDO Alliance’s role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment.

Table of Contents

1. Notation§

Type names, attribute names and element names are written as`code`

String literals are enclosed in “”, e.g. “UAF-TLV”.

In formulas we use “|” to denote byte wise concatenation operations.

The notation `base64url(byte[8..64])` reads as 8-64 bytes of data encoded in base64url, "Base 64 Encoding with URL and Filename Safe Alphabet" [RFC4648] *without padding*.

Following [WebIDL-ED], dictionary members are optional unless they are explicitly marked as`required`.

WebIDL dictionary members MUST NOT have a value of null.

Unless otherwise specified, if a WebIDL dictionary member is DOMString, it MUST NOT be empty.

Unless otherwise specified, if a WebIDL dictionary member is a List, it MUST NOT be an empty list.

For definitions of terms, please refer to the FIDO Glossary[FIDOGlossary].

All diagrams, examples, notes in this specification are non-normative.

Note: Certain dictionary members need to be present in order to comply with FIDO requirements. Such members are marked in the WebIDL definitions found in this document, as `required`. The keyword `required` has been introduced by [WebIDL-ED], which is a work-in-progress. If you are using a WebIDL parser which implements [WebIDL], then you may remove the keyword `required` from your WebIDL and use other means to ensure those fields are present.

1.1. Key Words§

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

2. Overview§

This section is not normative.

[FIDOMetadataStatement] defines authenticator metadata statements.

These metadata statements contain the trust anchor required to verify the attestation object (more specifically the `KeyRegistrationData` object), and they also describe several other important characteristics of the authenticator, including supported authentication and registration assertion schemes, and key protection flags.

These characteristics can be used when defining policies about which authenticators are acceptable for registration or authentication.

The metadata service described in this document defines a baseline method for relying parties to access the latest metadata statements.

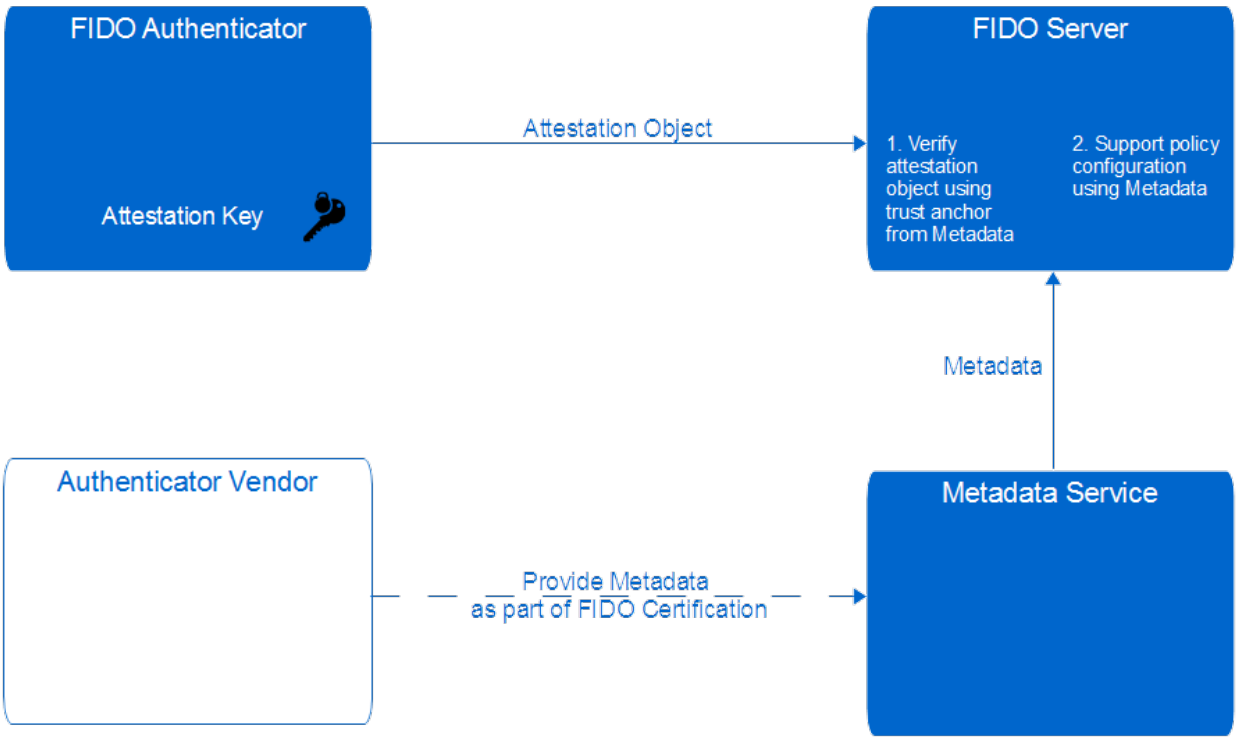

Figure 1 *FIDO Metadata Service Architecture Overview*

2.1. Scope§

This document describes the FIDO Metadata Service architecture in detail and it defines the structure and

interface to access this service. It also defines the flow of the metadata related messages and presents the rationale behind the design choices.

2.2. Detailed Architecture§

The metadata BLOB file contains a list of metadata statements related to the authenticators known to the FIDO Alliance (FIDO Authenticators).

The FIDO Server downloads the metadata BLOB file from a well-known FIDO URL and caches it locally.

The FIDO Server verifies the integrity and authenticity of this metadata BLOB file using the digital signature. It then iterates through the individual entries and parses the metadata statements related to authenticator models relevant to the relying party.

Individual metadata statements are included in the entry of the metadata BLOB file, and may be cached by the FIDO Server as required.

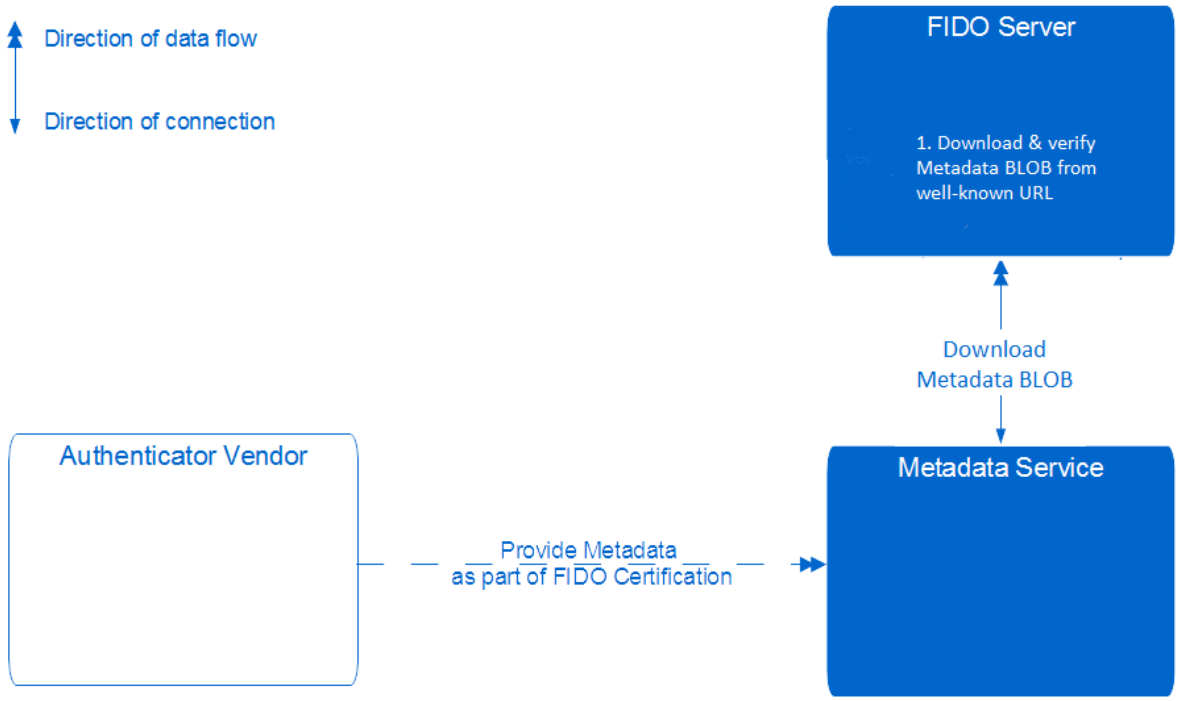

Figure 2 *FIDO Metadata Service Architecture*

> The single arrow indicates the direction of the network connection, the double arrow indicates the direction of the data flow.

> The metadata BLOB file is accessible at a well-known URL published by the FIDO Alliance.

> The relying party decides how frequently the metadata service is accessed to check for metadata BLOB updates.

3. Metadata Service Details§

This section is normative.

The relying party can decide whether it wants to use the metadata service and whether or not it wants to accept certain authenticators for registration or authentication.

The relying party could also obtain metadata directly from authenticator vendors or other trusted sources.

3.1. Metadata BLOB Format§

The metadata service makes the metadata BLOB object (see Metadata BLOB) accessible to FIDO Servers.

This object contains all metadata for each authenticator including the metadata statements defined in[FIDOMetadataStatement]. The BLOB object contains one signature.

3.1.1. Metadata BLOB Payload Entry dictionary§

Represents the MetadataBLOBPayloadEntry

```
dictionary MetadataBLOBPayloadEntry {
    AAID                    aaid;
    AAGUID                  aaguid;
    DOMString[]             attestationCertificateKeyIdentifiers;
    MetadataStatement       metadataStatement;
    BiometricStatusReport[] biometricStatusReports;
    required StatusReport[] statusReports;
    required DOMString      timeOfLastStatusChange;
    DOMString               rogueListURL;
    DOMString               rogueListHash;
};
```

`aaid`, of type AAID
: The AAID of the authenticator this metadata BLOB payload entry relates to. See[UAFProtocol] for the definition of the AAID structure. This field MUST be set if the authenticator implements FIDO UAF.

Note: FIDO UAF authenticators support AAID, but they don't support AAGUID.

`aaguid`, of type AAGUID
: The Authenticator Attestation GUID. See[FIDOKeyAttestation] for the definition of the AAGUID structure. This field MUST be set if the authenticator implements FIDO2.

Note: FIDO2 authenticators support AAGUID, but they don't support AAID.

`attestationCertificateKeyIdentifiers`, of type DOMString[]
: A list of the attestation certificate public key identifiers encoded as hex string. This value MUST be calculated according to method 1 for computing the keyIdentifier as defined in [RFC5280] section 4.2.1.2.

- The hex string MUST NOT contain any non-hex characters (e.g. spaces).
- All hex letters MUST be lower case.
- This field MUST be set if neither`aaid` nor `aaguid` are set. Setting this field implies that the attestation certificate(s) are dedicated to a single authenticator model.

FIDO U2F authenticators do not support AAID nor AAGUID, but they use attestation certificates dedicated to a single authenticator model.

`metadataStatement`, of type MetadataStatement
: The metadataStatement JSON object as defined in [FIDOMetadataStatement].

`biometricStatusReports`, of type BiometricStatusReport[]
: Status of the FIDO Biometric Certification of one or more biometric components of the Authenticator [FIDOBiometricsRequirements].

`statusReports`, of type StatusReport[]
: An array of status reports applicable to this authenticator.

`timeOfLastStatusChange`, of type DOMString
: ISO-8601 formatted date since when the status report array was set to the current value.

`rogueListURL`, of type DOMString
: URL of a list of rogue (i.e. untrusted) individual authenticators.

`rogueListHash`, of type DOMString
: `base64url(string[1..512])`

The hash value computed over the Base64url encoding of the UTF-8 representation of the JSON encoded rogueList available at `rogueListURL` (with type rogueListEntry[]). The hash algorithm related to the signature algorithm specified in the JWTHeader (see Metadata BLOB) MUST be used.

This hash value MUST be present and non-empty whenever `rogueListURL` is present.

> This method of base64url-encoding the UTF-8 representation is also used by JWT [JWT] to avoid encoding ambiguities.

EXAMPLE 1

```
{
  "no": 1234,
  "nextUpdate": "2014-03-31",
  "entries": [
    {
      "aaid": "1234#5678",
      "metadataStatement": "Metadata Statement object as defined in Metadata Statement spec."
,
      "statusReports": [
        {
          "status": "FIDO_CERTIFIED",
          "effectiveDate": "2014-01-04"
        }
      ],
      "timeOfLastStatusChange": "2014-01-04"
    },
    {
      "attestationCertificateKeyIdentifiers": [
        "7c0903708b87115b0b422def3138c3c864e44573"
      ],
      "metadataStatement": "Metadata Statement object as defined in Metadata Statement spec."
,
      "statusReports": [
        {
          "status": "FIDO_CERTIFIED",
          "effectiveDate": "2014-01-07"
        },
        {
          "status": "UPDATE_AVAILABLE",
          "effectiveDate": "2014-02-19",
          "url": "https://example.com/update1234"
        }
      ],
      "timeOfLastStatusChange": "2014-02-19"
    }
  ]
}
```

3.1.2. BiometricStatusReport dictionary§

Contains the current `BiometricStatusReport` of one of the authenticator's biometric component.

```
dictionary BiometricStatusReport {
    required unsigned short certLevel;
    required DOMString      modality;
    DOMString               effectiveDate;
    DOMString               certificationDescriptor;
    DOMString               certificateNumber;
    DOMString               certificationPolicyVersion;
    DOMString               certificationRequirementsVersion;
};
```

`certLevel`, of type unsigned short
: Achieved level of the biometric certification of this biometric component of the authenticator [FIDOBiometrics Requirements].

`modality`, of type DOMString
: A *single* a single USER_VERIFY short form case-sensitive string name constant, representing biometric

modality. See section "User Verification Methods" in [FIDORegistry] (e.g. "fingerprint_internal"). This value MUST NOT be empty and this value MUST correspond to one or more entries in field `userVerificationDetails` in the related Metadata Statement [FIDOMetadataStatement]. This value MUST represent a biometric modality.

> For example use `USER_VERIFY_FINGERPRINT` for the fingerprint based biometric component. In this case the related Metadata Statement must also claim fingerprint as one of the user verification methods.

`effectiveDate`, of type DOMString
ISO-8601 formatted date since when the `certLevel` achieved, if applicable. If no date is given, the status is assumed to be effective while present.

`certificationDescriptor`, of type DOMString
Describes the externally visible aspects of the Biometric Certification evaluation.

> For example it could state that the "biometric component is implemented OnChip - keeping biometric data inside the chip only.".

`certificateNumber`, of type DOMString
The unique identifier for the issued Biometric Certification.

`certificationPolicyVersion`, of type DOMString
The version of the Biometric Certification Policy the implementation is Certified to, e.g. "1.0.0".

`certificationRequirementsVersion`, of type DOMString
The version of the Biometric Requirements [FIDOBiometricsRequirements] the implementation is certified to, e.g. "1.0.0".

3.1.3. StatusReport dictionary§

> Contains an `AuthenticatorStatus` and additional data associated with it, if any.
>
> New `StatusReport` entries will be added to report known issues present in firmware updates.

The latest `StatusReport` entry MUST reflect the "current" status. For example, if the latest entry has status `USER_VERIFICATION_BYPASS`, then it is recommended assuming an increased risk associated with all authenticators of this AAID; if the latest entry has status `UPDATE_AVAILABLE`, then the update is intended to address at least all previous issues *reported* in this StatusReport dictionary.

```
dictionary StatusReport {
    required AuthenticatorStatus status;
    DOMString                    effectiveDate;
    unsigned long                authenticatorVersion;
    DOMString                    certificate;
    DOMString                    url;
    DOMString                    certificationDescriptor;
    DOMString                    certificateNumber;
    DOMString                    certificationPolicyVersion;
    DOMString                    certificationRequirementsVersion;
};
```

`status`, of type AuthenticatorStatus
Status of the authenticator. Additional fields MAY be set depending on this value.

`effectiveDate`, of type DOMString
ISO-8601 formatted date since when the status code was set, if applicable. If no date is given, the status is assumed to be effective while present.

`authenticatorVersion`, of type unsigned long

The authenticatorVersion that this status report relates to. In the case of FIDO_CERTIFIED* status values, the status applies to higher authenticatorVersions until there is a new statusReport.

For example, if the status would be USER_VERIFICATION_BYPASS, the authenticatorVersion indicates the vulnerable firmware version of the authenticator. Similarly, if the status would be UPDATE_AVAILABLE, the authenticatorVersion indicates the updated firmware version that is available now. If the status would be SELF_ASSERTION_SUBMITTED, the authenticatorVersion indicates the firmware version that the self assertion was based on.

`certificate`, of type DOMString
Base64-encoded [RFC4648] (not base64url!) DER [ITU-X690-2008] PKIX certificate value related to the current status, if applicable.

As an example, this could be an Attestation Root Certificate (see[FIDOMetadataStatement]) related to a set of compromised authenticators (ATTESTATION_KEY_COMPROMISE).

`url`, of type DOMString
HTTPS URL where additional information may be found related to the current status, if applicable.

For example a link to a web page describing an available firmware update in the case of status `UPDATE_AVAILABLE`, or a link to a description of an identified issue in the case of status `USER_VERIFICATION_BYPASS`.

`certificationDescriptor`, of type DOMString
Describes the externally visible aspects of the Authenticator Certification evaluation.

For example it could state that the authenticator is a "SecurityKey based on a CC EAL 5 certified chip hardware".

`certificateNumber`, of type DOMString
The unique identifier for the issued Certification.

`certificationPolicyVersion`, of type DOMString
The version of the Authenticator Certification Policy the implementation is Certified to, e.g. "1.0.0".

`certificationRequirementsVersion`, of type DOMString
The Document Version of the Authenticator Security Requirements (DV)[FIDOAuthenticatorSecurityRequirements] the implementation is certified to, e.g. "1.2.0".

3.1.4. AuthenticatorStatus enum§

This enumeration describes the status of an authenticator model as identified by its AAID/AAGUID or attestationCertificateKeyIdentifiers and potentially some additional information (such as a specific attestation key).

```
enum AuthenticatorStatus {
    "NOT_FIDO_CERTIFIED",
    "FIDO_CERTIFIED",
    "USER_VERIFICATION_BYPASS",
    "ATTESTATION_KEY_COMPROMISE",
    "USER_KEY_REMOTE_COMPROMISE",
    "USER_KEY_PHYSICAL_COMPROMISE",
    "UPDATE_AVAILABLE",
    "REVOKED",
    "SELF_ASSERTION_SUBMITTED",
    "FIDO_CERTIFIED_L1",
    "FIDO_CERTIFIED_L1plus",
    "FIDO_CERTIFIED_L2",
    "FIDO_CERTIFIED_L2plus",
    "FIDO_CERTIFIED_L3",
    "FIDO_CERTIFIED_L3plus"
};
```

3.1.4.1. Certification Related Statuses

NOT_FIDO_CERTIFIED
This authenticator is not FIDO certified.

> Applicable StatusReport fields are:

- effectiveDate - When status was achieved
- authenticatorVersion - The minimum applicable authenticator version.
- url - To the authenticator page or additional information about the authenticator

SELF_ASSERTION_SUBMITTED
The authenticator vendor has completed and submitted the self-certification checklist to the FIDO Alliance. If this completed checklist is publicly available, the URL will be specified in url.

> Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - New authenticator version that is

FIDO_CERTIFIED
This authenticator has passed FIDO functional certification. This certification scheme is phased out and will be replaced by FIDO_CERTIFIED_L1.

> Applicable StatusReport fields are:

- effectiveDate - When certification was issued
- authenticatorVersion - The minimum version of the certified solution
- certificationDescriptor - Authenticator Description. I.e. "Munikey 7c Black Edition"
- certificateNumber - FIDO Alliance Certificate Number
- certificationPolicyVersion - Authenticator Certification Policy
- certificationRequirementsVersion - Security Requirements Version
- url - URL to the certificate, or the news article about achievement of the certification.

These fields are applicable to any of the FIDO_CERTIFIED_*.

FIDO_CERTIFIED_L1
The authenticator has passed FIDO Authenticator certification at level 1. This level is the more strict successor of FIDO_CERTIFIED.

FIDO_CERTIFIED_L1plus
The authenticator has passed FIDO Authenticator certification at level 1+. This level is the more than level 1.

FIDO_CERTIFIED_L2
The authenticator has passed FIDO Authenticator certification at level 2. This level is more strict than level 1+.

FIDO_CERTIFIED_L2plus
The authenticator has passed FIDO Authenticator certification at level 2+. This level is more strict than level 2.

FIDO_CERTIFIED_L3
The authenticator has passed FIDO Authenticator certification at level 3. This level is more strict than level 2+.

FIDO_CERTIFIED_L3plus
The authenticator has passed FIDO Authenticator certification at level 3+. This level is more strict than level 3.

REVOKED
The FIDO Alliance has determined that this authenticator should not be trusted for any reason. For example if it is known to be a fraudulent product or contain a deliberate backdoor. Relying parties SHOULD reject any future registration of this authenticator model.

> Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - New authenticator version that is
- url - URL to the news/corporate article explaining the reason for revocation

3.1.4.2. Security Notification Statuses

USER_VERIFICATION_BYPASS
Indicates that malware is able to bypass the user verification. This means that the authenticator could be used without the user's consent and potentially even without the user's knowledge.

> Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - Minimum affected authenticator version
- url - URL to the news/corporate article explaining the incident

ATTESTATION_KEY_COMPROMISE
Indicates that an attestation key for this authenticator is known to be compromised. The relying party SHOULD check the certificate field and use it to identify the compromised authenticator batch. If the certificate field is not set, the relying party should reject all new registrations of the compromised authenticator. The Authenticator manufacturer should set the date to the date when compromise has occurred.

> Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - Minimum affected authenticator version

- certificate - Base64 DER-encoded PKIX certificate identifying compromised attestation root. If missing, then assume all authenticators of this model are compromised.
- url - URL to the news/corporate article explaining the incident

USER_KEY_REMOTE_COMPROMISE
This authenticator has identified weaknesses that allow registered keys to be compromised and should not be trusted. This would include both, e.g. weak entropy that causes predictable keys to be generated or side channels that allow keys or signatures to be forged, guessed or extracted.

> Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - Minimum affected authenticator version
- url - URL to the news/corporate article explaining the incident

USER_KEY_PHYSICAL_COMPROMISE
This authenticator has known weaknesses in its key protection mechanism(s) that allow user keys to be extracted by an adversary in physical possession of the device.

> Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - Minimum affected authenticator version
- url - URL to the news/corporate article explaining the incident

3.1.4.3. Info Statuses

UPDATE_AVAILABLE
A software or firmware update is available for the device. The Authenticator manufacturer should set the url to the URL where users can obtain an update and the date the update was published. When this status code is used, then the field authenticatorVersion in the authenticator Metadata Statement [FIDOMetadataStatement] MUST be updated, if the update fixes severe security issues, e.g. the ones reported by preceding StatusReport entries with status code USER_VERIFICATION_BYPASS, ATTESTATION_KEY_COMPROMISE, USER_KEY_REMOTE_COMPROMISE, USER_KEY_PHYSICAL_COMPROMISE, REVOKED. The Relying party MUST reject the Metadata Statement if the authenticatorVersion has not increased

> Applicable StatusReport fields are:

- effectiveDate - Date of incident being reported
- authenticatorVersion - New authenticator version that is available. MUST match authenticatorVersion in the metadata statement.
- url - URL to the page with the update info

> Relying parties might want to inform users about available firmware updates.

More values might be added in the future. FIDO Servers MUST silently ignore all unknown AuthenticatorStatus values.

3.1.5. RogueListEntry dictionary

Contains a list of individual authenticators known to be rogue.
New `RogueListEntry` entries will be added to report new individual authenticators known to be rogue.

Old `RogueListEntry` entries will be removed if the individual authenticator is known to not be rogue any longer.

Contains a list of individual authenticators known to be rogue.

New `RogueListEntry` entries will be added to report new individual authenticators known to be rogue.

Old `RogueListEntry` entries will be removed if the individual authenticator is known to not be rogue any longer.

```
dictionary RogueListEntry {
    required DOMString sk;
    required DOMString date;
};
```

`sk`, of type DOMString
: Base64url encoding of the rogue authenticator's secret key (sk value, see[FIDOEcdaaAlgorithm], section ECDAA Attestation).

In order to revoke an individual authenticator, its secret key (sk) must be known.

`date`, of type DOMString
: ISO-8601 formatted date since when this entry is effective.

```
EXAMPLE: ROGUELISTENTRY[] EXAMPLE
[
    { "sk": "MO-oaqbeJSSayzXaDUhh9LMKeT4Zio1bqn6W8kDaUfM",
      "date": "2016-06-07"},
    { "sk": "k96Npt4jJIq7NNoNSGH0swp5PhU6jVuyf5jyYNtxrNQ",
      "date": "2016-06-09"},
]
```

3.1.6. Metadata BLOB Payload dictionary

Represents the MetadataBLOBPayload

```
dictionary MetadataBLOBPayload {
    DOMString                          legalHeader;
    required Number                    no;
    required DOMString                 nextUpdate;
    required MetadataBLOBPayloadEntry[] entries;
};
```

`legalHeader`, of type DOMString
: The legalHeader, which MUST be in each BLOB, is an indication of the acceptance of the relevant legal agreement for using the MDS. The FIDO Alliance's Blob will contain this legal header: "legalHeader": "Retrieval and use of this BLOB indicates acceptance of the appropriate agreement located at https://fidoalliance.org/metadata/metadata-legal-terms/"

`no`, of type Number
: The serial number of this UAF Metadata BLOB Payload. Serial numbers MUST be consecutive and strictly monotonic, i.e. the successor BLOB will have a `no` value exactly incremented by one.

`nextUpdate`, of type DOMString
: ISO-8601 formatted date when the next update will be provided at latest.

`entries`, of type MetadataBLOBPayloadEntry[]
List of zero or more MetadataBLOBPayloadEntry objects.

3.1.7. Metadata BLOB

The metadata BLOB is a JSON Web Token (see [JWT] and [JWS]). It consists of three elements:

- The base64url encoding, without padding, of the UTF-8 encoded JWT Header (see example below),
- the base64url encoding, without padding, of the UTF-8 encoded Metadata BLOB Payload (see example at the beginning of section Metadata BLOB Format>),
- and the base64url-encoded, also without padding, JWS Signature [JWS] computed over the to-be-signed payload using the Metadata BLOB signing key, i.e. `tbsPayload = EncodedJWTHeader | "." | EncodedMetadataBLOBPayload`

All three elements of the BLOB are concatenated by a period ("."): `MetadataBLOB = EncodedJWTHeader | "." | EncodedMetadataBLOBPayload | "." | EncodedJWSSignature`

The hash algorithm related to the signing algorithm specified in the JWT Header (e.g. SHA256 in the case of "ES256") MUST also be used to compute the hash of the metadata statements (see section Metadata BLOB Payload Entry Dictionary).

3.1.7.1. Examples

This section is not normative.

```
EXAMPLE: ENCODED METADATA BLOB
ewoJImxlZ2FsSGVhZGVyIjogIlJldHJpZXZhbCBhbmQgdXNlIG9mIHRoaXMgQkxPQiBpbmRpY2F0ZXMg
YWNjZXB0YW5jZSBvZiB0aGUgYXBwcm9wcmlhdGUgYWdyZWVtZW50IGxvY2F0ZWQgYXQgaHR0cHM6Ly9m
aWRvYWxsaWFuY2Uub3JnL21ldGFkYXRhL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsCgkibm8iOiAxNSwK
CSJuZXh0VXBkYXRlIjogIjIwMjAtMDMtMzAiLAoJImVudHJpZXMiOiBbewoJCQkiYWFpZCI6ICIxMjM0
IzU2NzgiLAoJCQkibWV0YWRhdGFTdGF0ZW1lbnQiOiB7CgkJCQkibGVnYWxIZWFkZXIiOiAiaHR0cHM6
Ly9maWRvYWxsaWFuY2Uub3JnL21ldGFkYXRhL21ldGFkYXRhLXN0YXRlbWVudC1sZWdhbC1oZWFkZXIv
IiwKCQkJCSJkZXNjcmlwdGlvbiI6ICJGSURPIEFsbGlhbmNlIFNhbXBsZSBVQUYgQXV0aGVudGljYXRv
ciIsCgkJCQkiYWFpZCI6ICIxMjM0IzU2NzgiLAoJCQkJImFsdGVybmF0aXZlRGVzY3JpcHRpb25zIjog
ewoJCQkJCSJydS1SVSI6ICLQn9GA0LjQvNC10YAgVUFGINCw0YPRgtC10L3RgtC40YTQuNC60LDRgtC-
0YDQsCDQvtGCIEZJRE8gQWxsaWFuY2UiLAoJCQkJCSJmci1GUiI6ICJFeGVtcGxlIFVBRiBhdXRoZW50
aWNhdG9yIGRlIEZJRE8gQWxsaWFuY2UiCgkJCQl9LAoJCQkJImF1dGhlbnRpY2F0b3JWZXJzaW9uIjog
MiwKCQkJCSJwcm90b2NvbEZhbWlseSI6ICJ1YWYiLAoJCQkJInNjaGVtYSI6IDMsCgkJCQkidXB2Ijog
W3sKCQkJCQkJIm1ham9yIjogMSwKCQkJCQkJIm1pbm9yIjogMAoJCQkJCX0sCgkJCQkJewoJCQkJCQki
bWFqb3IiOiAxLAoJCQkJCQkibWlub3IiOiAxCgkJCQkJfQoJCQkJXSwKCQkJCSJhdXRoZW50aWNhdGlv
bkFsZ29yaXRobXMiOiBbInNlY3AyNTZyMV9lY2RzYV9zaGEyNTZfcmF3Il0sCgkJCQkicHVibGljS2V5
QWxnQW5kRW5jb2RpbmdzIjogWyJlY2NfeDk2Ml9yYXciXSwKCQkJCSJhdHRlc3RhdGlvblR5cGVzIjog
WyJiYXNpY19mdWxsIl0sCgkJCQkidXNlclZlcmlmaWNhdGlvbkRldGFpbHMiOiBbCgkJCQkJW3sKCQkJ
CQkJInVzZXJWZXJpZmljYXRpb25NZXRob2QiOiAiZmluZ2VycHJpbnRfaW50ZXJuYWwiLAoJCQkJCQki
YmFEZXNjIjogewoJCQkJCQkJInNlbGZBdHRlc3RlZEBUiI6IDAuMDAwMDIsCgkJCQkJCQkibWF4UmV0
cmllcyI6IDUsCgkJCQkJCQkiYmxvY2tTbG93ZG93biI6IDMwLAoJCQkJCQkJIm1heFRlbXBsYXRlcyI6
IDUKCQkJCQkJfQoJCQkJCX1dCgkJCQldLAoJCQkJImtleVByb3RlY3Rpb24iOiBbImhhcmR3YXJlIiwg
InRlZSJdLAoJCQkJImlzS2V5UmVzdHJpY3RlZCI6IHRydWUsCgkJCQkibWF0Y2hlclByb3RlY3Rpb24i
OiBbInRlZSJdLAoJCQkJImNyeXB0b1N0cmVuZ3RoIjogMTI4LAoJCQkJImF0dGFjaG1lbnRIaW50Ijog
WyJpbnRlcm5hbCJdLAoJCQkJInRjRGlzcGxheSI6IFsiYW55IiwgInRlZSJdLAoJCQkJInRjRGlzcGxh
eUNvbnRlbnRUeXBlIjogImltYWdlL3BuZyIsCgkJCQkidGNEaXNwbGF5UE5HQ2hhcmFjdGVyaXN0aWNz
IjogW3sKCQkJCQkid2lkdGgiOiAzMjAsCgkJCQkJImhlaWdodCI6IDQ4MCwKCQkJCQkiYml0RGVwdGgi
OiAxNiwKCQkJCQkiY29sb3JUeXBlIjogMiwKCQkJCQkiY29tcHJlc3Npb24iOiAwLAoJCQkJCSJmaWx0
ZXIiOiAwLAoJCQkJCSJpbnRlcmxhY2UiOiAwCgkJCQl9XSwKCQkJCSJhdHRlc3RhdGlvblJvb3RDZXJ0
aWZpY2F0ZXMiOiBbCgkJCQkJIk1JSUNQVENDQWVPZ0F3SUJBZ0lKQU91ZXh2VTNPeTJ3TUFvR0NDcUdT
TTQ5QkFNQ01Ic3hJREFlQmdOVkJBTU1GMU5oYlhCc1pTQkJkSFJsYzNSaGRHbHZiaUJTYjI5ME1SWXdG
QVlEVlFRS0RBMUdTVVJQSUVGc2JHbGhibU5sTVJFd0R3WURWUVFMREFoVlFVWWdWRmRITERFU01CQUdB
MVVFQnd3SlVHRnNieUJCYkhSdk1Rc3dDUVlEVlFRSURBSkRRVEVMTUFrR0ExVUVCaE1DVlZNd0hoY05N
```

```
VFF3TmpFNE1UTXpNek15V2hjTk5ERXhNVEF6TVRNek16TXlXakI3TVNBd0hnWURWUVFEREJkVFlXMXdi
R1VnUVhSMFpYTjBZWFJwYjI0Z1VtOXZkREVXTUJRR0ExVUVDZ3dOUmtsRVR5QkJiR3hwWVc1alpURVJN
QThHQTFVRUN3d0lWVUZHSUZSWFJ5d3hFakFRQmdOVkJBY01DVkJoYkc4Z1FXeDBiekVMTUFrR0ExVUVD
QXdDUTBFeEN6QUpCZ05WQkFZVEFsVlRNRmt3RXdZSEtvWkl6ajBDQVFZSUtvWkl6ajBEQVFjRFFnQUVI
OGh2MkQwSFhhNTkvQm1wUTdSWmVoTC9GTUd6RmQxUUJnOXZBVXBPWjNham51UTk0UFI3YU16SDMzblVT
QnI4ZkhZRHJxT0JiNThweEdxSEpSeVgvNk5RTUU0d0hRWURWUjBPQkJZRUZQb0hBM0NMaHhGYkMwSXQ3
ekU0dzhoazVFSi9NQjhHQTFVZEl3UVlNQmFBRlBvSEEzQ0xoeEZiQzBJdDd6RTR3OGhrNUVKL01Bd0dB
MVVkRXdRRk1BTUJBZjh3Q2dZSUtvWkl6ajBFQXdJRFNBQXdSUUloQUowNlFTWHQ5aWhJYkVLWUtJanNQ
a3JpVmRMSWd0ZnNiRFN1N0VySmZ6cjRBaUJxb1lDWmYwK3pJNTVhUWVBSGpJekE5WG02M3JydUF4Qlo5
cHM5ejJYTmxRPT0iCgkJCQldLAoJCQkJImljb24iOiAiZGF0YTppbWFnZS9wbmc7YmFzZTY0LGlWQk9S
dzBLR2dvQUFBQU5TVWhFVWdBQUFFOEFBQUF2Q0FZQUFBQ2l3SmZjQUFBQUFYTlNSMElBcnM0YzZRQUFB
QVJuUVUxQkFBQ3hqd3Y4WVFVQUFBQUpjRWhaY3dBQURzTUFBQTdEQWNkdnFHUUFBQWFoU1VSQlZHaEQ3
WnI1YnhSbEdNZjlLelRCOEFNL1lFaEUyVzdwUVpjV0tLQmNsU3BIQVRsRUxBUkU3a05FQ0NBM0ZrV0sw
Q0tLU0NGSXNLQmNnVkNEV0dORVNkQVlpZHdnZ2dKQmlSaU1oRmMvNHd5ODg4NHp1OU5kbG5HVGZaSlAy
bjNuTysrODg5MzNmdmVCQngrUHFDekprVFV2QmJMbXBVRFd2QlRJbXBjQ1NadlhMQ2RYOVIwNVNrMTli
YjVhdGY1OTlmRysvZXJBNTQxcTQ3YVAxTExWYTlTSXlWTlVpOElpOGQ1a0dUc2kzME5Gdjdha TluN1Fa
UE13YmR5czJlclUyWE1xVWR5OCtaY2FObUdpbUU4eVhOM1JVZDNhMThuRjBmVWxvdlorMENUeldwZDJW
aitlT20xYkV5eTZEeDRpNXBVTUdXdmVvNTA2cTIyN2R0dVdCSXVmZnI2b1dwVjBGUE5MaG93MTc1MU5t
MjFMdlBIM3JWdFdqZno2NkxmcWw4dFg3RlJsOVlGU1hzbVNzZWI5Y2VPR2JZazdNTlVjR1BnOFpzYk1l
OXJmUVVhYVYvSk1YOXNxZHpEQ1N2cDBrWkhtVFpnOXg3YkxIY01uVGhiMTZlSittVmZRcTh5YVVaUU5H
NjRpWForMC9rcTZ1T1pGTzBRdGF0ZFdLZlhuUlE5OUJqOTFSNU9JRm5rNTRqTjBta1VpcWxPM1hEVytN
bCs5OG1LQjZ0VzdyV3BaY1BjKzB6ZzR0THJZbFVjODZFNmVHRGpJTXViVnBjdXNlYXJmZ0lZR1JrNmJy
aFpWci9KY0h6b29MNzU1MGplZExFeG9wV2NBcGkyWlVxaHU3Skx2clZzUVU4MXprek9QZWVtTVJZdlZ1
UXNYN1BiaURRWTVKdlpvbmZ0SysxVlk4SDl1dHg1MzBoMG9iK2ptUllxajZvdWFZdkVlblcvV2xZanA4
Y3diTW02ODJ0UHdxVzFSNHRqLzJTSDEzSVJKWWw0bW9adlhwaVNxRHI3ZFh0UUh4YS9QSzMvK0JXc0sx
ZFRnSHU2Vjh0UUozYndGa3dwRnJVT1E1MHMxcjNsZXZtOHpaY3ExNytCQmF3N0s4bEVLNXF6a1llYXJr
OUE4cDdQM0d6REsrbmQzRFFvdys2VUM4U1ZOODJpdXYzOGltN050YVh0VjFDVnE2Umd3NHBrc21iZGkz
YnUyRGU3WWZhQkJ4Y3FmdnFQclVqRlFOVFEyMmxmZFVWVlQ2OHJUSktGNURuU21VamdkcWc0bVNTOXBt
c2ZESlIzRzZUb0gwaVc5YVY3TFdMSFlYS2xsVER0MExUQXRrWUlhYW1wMVFqVnYrK3V5R1V4VmRKMERO
VlhTbStiMXFSeHBsODRkZGZYMUxwMU8vZDY5dHNvZDB2czVoR3JlOXh1OG8rZnBMUjFjR2hOVEQ2WjU3
QzlLTVdYZWZKZE9aOTRiYjlvcWQxUk9uUzdxSVRUekhpbU1xaXZiTzNnMERkVnlrM1dRQmhCenRLMzVZ
S05kT25jOE8zYWNTNmZEWkZnS2FYTHNFSnA1cmRybGlCcXA4OWNKY3MvbTdUdnMwcmtqR2ZONGIwa1Bv
Wm4zVUp1SU9ybloyMnlQMWZtdlV4K081Z1NxZWJWMW0relN1WU5WaHE3VFdiRGlMVnZsanBsTGxvcDZD
TFhQKzJxdHZHTElMLzF2aW1JU2RNQmd6U29GWnl1NlRxZCtqenhnc1BhVjlCQ3FlZS9OallrNnY2bEs5
Y3dpVWMvU1R0ZjFIRHBNM2I1OTJ5N2gzVGh4NW96SzY5SExwWVd1QXdhcVM1Y3YyNnE3Y2ViOGVmVllh
UmVQM2lGVTh6ajFrblN3WlhITW1uQ2pZME9nYWxvN1VRZlNDTTNxUVFyMkgvWEZQN3NzWHg0NVlsOTFC
eWVDZXA0bW9ab0grMWZHM3hENHRUN3g4a3d5ajhud2I5ZXYyNlYwQjZkKzdINHpLdnVkQUg1MzdGanF5
ek9IZEpuSEV1em1YcS9XanhPYnZOTWJ2N25oeXdzWDJhVnNXdEM4KzQ4YUxlYXBFN3A1d0taaTBBMkFR
UlY1bnZSNEUrdUpjK2I2MWtBcHFJbnhCZ21kLzRWNVFQL210MThIREM3c1JIZnRtZXU1bG1oVjBybi9B
TFgyMzJicWQ0QkZuRHg3VmkxY1dTMnVmZjBJYkI0N3FleHhtVWo5UXV0WWp1cGQzdFlENmFiV0JCTXJo
K2FwTmJPS3JORjErdWdDYTRyaVhHZndNUFB0VmlhdmhVM1lNT0FBbnVVYi9SMDdMMHlPU2VPYWRFODhB
cHNYRkdmZjMweW5obEpnTTUxQ1U2dk45RXpnbnB2SEJGVXlpVnJhZVBpd0o1M0RGNVpUWm5vbUVOZzg1
a05VZDJvSmkyV3ByNE9tbWtmTjR4NHpIZmlWRmM4RHY4Tnp1aE5xT2lkaWxHdkE2REd1ZVp3Tzc4QUFR
bjZjaUVrNitydzVWY3ZqdnFORFlQT29JVXdhS1NocnhBdVhMbGtINGFZdUdmTVlEYzEwV0Y1VGEzMWhQ
Sk9mY1VoclUvSmxJTmk2YzZlbFJZZEJwbzYrK1lmang2MWxHTmZSbTRNRDVySjFqM0ZvR0huakRTQk5h
cllVZ01MeU1zektwYjd0WHBvSGZQczhoM1dwMUx6TmZOazU0WHhDMXdER1VtWXpYWWVmaDZ6L2NLdFZt
NEVCeGE5VlFHRHpZcjNMclVNUmpIRUtrazd6YUZLWVFBMmhHUVUxeis4NU5GV3BYRHJrejN2eDEwR3F4
UTZCemVOYm9CazVuOGs0bmViUmgrazFoV2Z4VEYwRDFFeVdVczVuditkZ1FxS2F4enVDZEUwaXNIbDAy
TlE4YWgwbVhyMTJMYTNtMGY5d2lrOSt3TE5UTVkvODZNUG84eWkzMU9meG1UNlBXb3FHOStEWnVrWW5h
NTZtU1p0NVdXU3k1cVZBMXJ3VXlKcVhBbG56a2lhaS9nSFNEN1JrVHlpaG9nQUFBQUJKUlU1RXJrSmdn
Zz09IgoJCQl9LAoJCQkic3RhdHVzUmVwb3J0cyI6IFt7CgkJCQkic3RhdHVzIjogIkZJRE9fQ0VSVElG
SUVEIiwKCQkJCSJlZmZlY3RpdmVEYXRlIjogIjIwMTQtMDEtMDQiCgkJCX1dLAoJCQkidGltZU9mTGFz
dFN0YXR1c0NoYW5nZSI6ICIyMDE0LTAxLTA0IgoJCX0sCgkJewoJCQkiYWFndWlkIjogIjAxMzJkMTEw
LWJmNGUtNDIwOC1hNDAzLWFiNGY1ZjEyZWZlNSIsCgkJCSJtZXRhZGF0YVN0YXRlbWVudCI6IHsKCQkJ
CSJsZWdhbEhlYWRlciI6ICJodHRwczovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV0YWRhdGEt
c3RhdGVtZW50LWxlZ2FsLWhlYWRlci8iLAoJCQkJImRlc2NyaXB0aW9uIjogIkZJRE8gQWxsaWFuY2Ug
U2FtcGxlIEZJRE8yIEF1dGhlbnRpY2F0b3IiLAoJCQkJImFhZ3VpZCI6ICIwMTMyZDExMC1iZjRlLTQy
MDgtYTQwMy1hYjRmNWYxMmVmZTUiLAoJCQkJImFsdGVybmF0aXZlRGVzY3JpcHRpb25zIjogewoJCQkJ
CSJydS1SVSI6ICLQn9GA0LjQvNC10YAgRklETzIg0LDRg9GC0LXQvdGC0LjRhNC40LrQsNGC0L7RgNCw
INC-0YIgRklETyBBbGxpYW5jZSIsCgkJCQkImZyLUZSIjogIkV4ZW1wbGUgRklETzIgYXV0aGVudGlj
YXRvciBkZSBGSURPIEFsbGlhbmNlIiwKCQkJCQkiemgtQ04iOiAi5L6G6IeqRklETyBBbGxpYW5jZeea
hOekuuS-i0ZJRE8y6Lqr5Lu96amX6K2J5ZmoIgoJCQkJfSwKCQkJCSJwcm90b2NvbEZhbWlseSI6ICJm
```

aWRvMiIsCgkJCQkic2NoZW1hIjogMywKCQkJCSJhdXRoZW50aWNhdG9yVmVyc2lvbiI6IDUsCgkJCQki
dXB2IjogW3sKCQkJCQkibWFqb3IiOiAxLAoJCQkJCSJtaW5vciI6IDAKCQkJCX1dLAoJCQkJImF1dGhl
bnRpY2F0aW9uQWxnb3JpdGhtcyI6IFsic2VjcDI1NnIxX2VjZHNhX3NoYTI1Nl9yYXciLCAicnNhc3Nh
X3BrY3N2MTVfc2hhMjU2X3JhdyJdLAoJCQkJInB1YmxpY5ncyI6IFsiY29z
ZSJdLAoJCQkJImF0dGVzdGF0aW9uVHlwZXMiOiBbImJhc2ljX2Z1bGwiXSwKCQkJCSJ1c2VyVmVyaWZp
Y2F0aW9uRGV0YWlscyI6IFsKCQkJCQlbewoJCQkJCQkidXNlclZlcmlmaWNhdGlvbk1ldGhvZCI6ICJu
b25lIgoJCQkJCX1dLAoJCQkJCVt7CgkJCQkJCSJ1c2VyVmVyaWZpY2F0aW9uTWV0aG9kIjogInByZXNl
bmNlX2ludGVybmFsIgoJCQkJCX1dLAoJCQkJCVt7CgkJCQkJCSJ1c2VyVmVyaWZpY2F0aW9uTWV0aG9k
IjogInBhc3Njb2RlX2V4dGVybmFsIiwKCQkJCQkJImNhRGVzYyI6IHsKCQkJCQkJCSJiYXNlIjogMTAs
CgkJCQkJCQkibWluTGVuZ3RoIjogNAoJCQkJCQl9CgkJCQkJfV0sCgkJCQkJW3sKCQkJCQkJCSJ1c2Vy
VmVyaWZpY2F0aW9uTWV0aG9kIjogInBhc3Njb2RlX2V4dGVybmFsIiwKCQkJCQkJCSJjYURlc2MiOiB7
CgkJCQkJCQkJImJhc2UiOiAxMCwKCQkJCQkJCQkibWluTGVuZ3RoIjogNAoJCQkJCQkJfQoJCQkJCQl9
LAoJCQkJCQl7CgkJCQkJCQkidXNlclZlcmlmaWNhdGlvbk1ldGhvZCI6ICJwcmVzZW5jZV9pbnRlcm5h
bCIKCQkJCQkJfQoJCQkJCV0KCQkJCV0sCgkJCQkia2V5UHJvdGVjdGlvbiI6IFsia3FyZHdhcmUiLCAi
c2VjdXJlX2VsZW1lbnQiXSwKCQkJCSJtYXRjaGVyUHJvdGVjdGlvbiI6IFsib25fY2hpcCJdLAoJCQkJ
ImNyeXB0b1N0cmVuZ3RoIjogMTI4LAoJCQkJImF0dGFjaG1lbnRIaW50IjogWyJleHRlcm5hbCIsICJ3
aXJlZCIsICJ3aXJlbGVzcyIsICJuZmMiXSwKCQkJCSJ0Y0Rpc3BsYXkiOiBbXSwKCQkJCSJhdHRlc3Rh
dGlvblJvb3RDZXJ0aWZpY2F0ZXMiOiBbCgkJCQkJIk1JSUNQVENDQWVPZ0F3SUJBZ0lKQU91ZXh2VTNP
eTJ3TUFvR0NDcUdTTTQ5QkFNQ01Ic3hJREFlQmdOVkJBTU1GMU5oYlhCc1pTQkJkSFJsYzNSaGRHbHZi
aUJTYjI5ME1SWXdGQVlEVlFRS0RBMUdTVVJQSUVGc2JHbGhibU5sTVJFd0R3WURWUVFMREFoVlFVWWdW
RmRITERFU01CQUdBMVVFQnd3SlVHRnNieUJCYkhSdk1Rc3dDUVlEVlFRSURBSkRRVEVMTUFrR0ExVUVC
aE1DVlZNd0hoY05NVFF3TmpFNE1UTXpNek15V2hjTk5ERXhNVEF6TVRNek16TXlXakI3TVNBd0hnWURW
UVFEREJkVFlXMXdiR1VnUVhSMFpYTjBZWFJwYjI0Z1VtOXZkREVXTUJRR0ExVUVDZ3dOUmtsRVR5QkJi
R3hwWVc1alpURVJNQThHQTFVRUN3d0lWUZHSUZSWFJ5d3hFakFRQmdOVkJBY01DVkJoYkc4Z1FXeDBi
ekVMTUFrR0ExVUVDQXdDUTBFeEN6QUpCZ05WQkFZVEFsVlRNRmt3RXdZSEtvWkl6ajBDQVFZSUtvWkl6
ajBEQVFjRFFnQUVIOGh2MkQwSFhhNTkvQm1wUTdSWmVoTC9GTUd6RmQxUUJnOXZBVXBPWjNham51UTk0
UFI3YU16SDMzblVTQnI4ZkhZRHJxT0JiNThweEdxSEpSeVgvNk5RTUU0d0hRWURWUjBPQkJZRUZQb0hB
M0NMaHhGYkMwSXQ3ekU0dzhoazVFSi9NQjhHQTFVZEl3UVlNQmFBRlBvSEEzQ0xoeEZiQzBJdDd6RTR3
OGhrNUVKL01Bd0dBMVVkRXdRRk1BTUJBZjh3Q2dZSUtvWkl6ajBFQXdJRFNBQXdSUUloQUowNlFTWHQ5
aWhJYkVLWUtJanNQa3JpVmRMSWd0ZnNiRFN1N0VySmZ6cjRBaUJxb1lDWmYwK3pJNTVhUWVBSGpJekE5
WG02M3JydUF4Qlo5cHM5ejJYTmxRPT0iCgkJCQldLAoJCQkJImljb24iOiAiZGF0YTppbWFnZS9wbmc7
YmFzZTY0LGlWQk9SdzBLR2dvQUFBQU5TVWhFVWdBQUFFOEFBQUF2Q0FZQUFBQ2l3SmZjQUFBQUFYTlNS
MElBcnM0YzZRQUFBQVJuUVUxQkFBQ3hqd3Y4WVFVQUFBQUpjRWhaY3dBQURzTUFBQTdEQWNkdnFHUUFB
QWFoU1VSQlZHaEQ3WnI1YnhSbEdNZjlLelRCOEFNL1lFaEUyVzdwUVpjV0tLQmNsU3BIQVRsRUxBUkU3
a05FQ0NBM0ZrV0swQ0tLU0NGSXNLQmNnVkNEV0dORVNkQVlpZHdnZ2dKQmlSaU1oRmMvNHd5ODg4NHp1
OU5kbG5HVGZaSlAybjNuTysrODg5MzNmdmVCQngrUHFDekprVFV2QmJMbXBVRFd2QlRJbXBjQ1NadlhM
Q2RYOVIwNVNrMTliYjVhdGY1OTlmRysvZXJBNTQxcTQ3YVAxTExWYTlTSXlWTlVpOElpOGQ1a0dUc2kz
ME5GdjdhaTluN1FaUE13YmR5czJlclUyWE1xVWR5OCtaY2FObUdpbUU4eVhOM1JVZDNhMThuRjBmVWxv
dlorMENUeldwZDJWaitlT20xYkV5eTZEeDRpNXBVTUdXdmVvNTA2cTIyN2R0dVdCSXVmZnI2b1dwVjBG
UE5MaG93MTc1MU5tMjFMdlBIM3JWdFdqZno2NkxmcWw4dFg3RlJsOVlGU1hzbVNzZWI5Y2VPR2JZazdN
TlVjR1BnOFpzYk1lOXJmUVVhYVYvSk1YOXNxZHpEQ1N2cDBrWkhtVFpnOXg3YkxIY01uVGhiMTZlSitt
VmZRcTh5YVVaUU5HNjRpWForMC9rcTZ1T1pGTzBRdGF0ZFdLZlhuUlE5OUJqOTFSNU9JRm5rNTRqTjBt
a1VpcWxPM1hEVytNbCs5OG1LQjZ0VzdyV3BaY1BjKzB6ZzR0THJZbFVjODZFNmVHRGpJTXViVnBjdXNl
YXJmZ0lZR1JrNmJyaFpWci9KY0h6b29MNzU1MGplZExFeG9wV2NBcGkyWlVxaHU3Skx2clZzUVU4MXpr
ek9QZWVtTVJZdlZ1UXNYN1BiaURRWTVKdlpvbmZ0SysxVlk4SDl1dHg1MzBoMG9iK2ptUllxajZvdWFZ
dkVlblcvV2xZanA4Y3diTW02ODJ0UHdxVzFSNHRqLzJTSDEzSVJKWWw0bW9adlhwaVNxRHI3ZFh0UUh4
YS9QSzMvK0JXc0sxZFRnSHU2Vjh0UUozYndGa3dwRnJVT1E1MHMxcjNsZXZtOHpaY3ExNytCQmF3N0s4
bEVLNXF6a1llYXJrOUE4cDdQM0d6RErsrbmQzRFFvdys2VUM4U1ZOODJpdXYzOGltN050YVh0VjFDVnE2
Umd3NHBrc21iZGkzYnUyRGU3WWZhQkJ4Y3FmdnFQclVqRlFOVFEyMmxmZFVWVlQ2OHJUSktGNURuU21V
amdkcWc0bVNTOXBtc2ZESlIzRzZUb0gwaVc5YVY3TFdMSFlYS2xsVER0MExUQXRrWUlhYW1wMVFqVnYr
K3V5R1V4VmRKMEROVlhTbStiMXFSeHBsODRkZGZYMUxwMU8vZDY5dHNvZDB2czVoR3JlOXh1OG8rZnBM
UjFjR2hOVEQ2WjU3QzlLTVdYZWZKZE9aOTRiYjlvcWQxUk9uUzdxSVRUekhpbU1xaXZiTzNnMERkVnlr
M1dRQmhCenRLMzVZS05kT25jOE8zYWNTNmZEWkZnS2FYTHNFSnA1cmRybGlCcXA4OWNKY3MvbTdUdnMw
cmtqR2ZONGIwa1BvWm4zVUp1SU9ybloyMnlQMWZtdlV4K081Z1NxZWJWMW0relN1WU5WaHE3VFdiRGlM
VnZsanBsTGxvcDZDTFhQKzJxdHZHTElMLzF2aW1JU2RNQmd6U29GWnl1NlRxZCtqenhnc1BhVjlCQ3Fl
ZS9OallrNnY2bEs5Y3dpVWMvU1R0ZjFIRHBNM2I1OTJ5N2gzVGh4NW96SzY5SExwWVd1QXdhcVM1Y3Yy
NnE3Y2ViOGVmVllhUmVQM2lGVTh6ajFrblN3WlhITW1uQ2pZME9nYWxvN1VRZlNDTTNxUVFyMkgvWEZQ
N3NzWHg0NVlsOTFCeWVDZXA0bW9ab0grMWZHM3hENHRUN3g4a3d5ajhud2I5ZXYyNlYwQjZkKzdINHpL
dnVkQUg1MzdGanF5ek9IZEpuSEV1em1YcS9XanhPYnZOTWJ2N25oeXdzWDJhVnNXdEM4KzQ4YUxlYXBF
N3A1d0taaTBBMkFRUlY1bnZSNEUrdUpjK2I2MWtBcHFJbnhCZ21kLzRWNVFQL210MThIREM3c1JIZnRt
ZXU1bG1oVjBybi9BTFgyMzJicWQ0QkZuRHg3VmkxY1dTMnVmZjBJYkI0N3FleHhtVWo5UXV0WWp1cGQz
dFlENmFiV0JCTXJoK2FwTmJPS3JORjErdWdDYTRyaVhHZndNUFB0VmlhdmhVM1lNT0FBbnVVYi9SMDdM

```
MHlPU2VPYWRFODhBcHNYRkdmZjMweW5obEpnTTUxQ1U2dk45RXpnbnB2SEJGVXlpVnJhZVBpd0o1M0RG
NVpUWm5vbUVOZzg1a05VZDJvSmkyV3ByNE9tbWtmTjR4NHpIZmlWRmM4RHY4Tnp1aE5xT2lkaWxHdkE2
REd1ZVp3Tzc4QUFRbjZjaUVrNitydzVWY3ZqdnFORFlQT29JVXdhS1NocnhBdVhMbGtINGFZdUdmTVlE
YzEwV0Y1VGEzMWhQSk9mY1VoclUvSmxJTmk2YzZlbFJZZEJwbzYrK1lmang2MWxHTmZSbTRNRDVySjFq
M0ZvR0huakRTQk5hcllVZ01MeU1zektwYjd0WHBvSGZQczhoM1dwMUx6TmZOazU0WHhDMXdER1VtWXpY
WWVmaDZ6L2NLdFZtNEVCeGE5VlFHRHpZcjNMclVNUmpIRUtrazd6YUZLWVFBMmhHUVUxeis4NU5GV3BY
RHJrejN2eDEwR3F4UTZCemVOYm9CazVuOGs0bmViUmgrazFoV2Z4VEYwRDFFeVdVczVuditkZ1FxS2F4
enVDZEUwaXNIbDAyTlE4YWgwbVhyMTJMYTNtMGY5d2lrOSt3TE5UTVkvODZNUG84eWkzMU9meG1UNlBX
b3FHOStEWnVrWW5hNTZtU1p0NVdXU3k1cVZBMXJ3VXlKcVhBbG56a2lhaS9nSFNEN1JrVHlpaG9nQUFB
QUJKUlU1RXJrSmdnZz09IiwKCQkJCSJzdXBwb3J0ZWRFeHRlbnNpb25zIjogW3sKCQkJCQkJImlkIjog
ImhtYWMtc2VjcmV0IiwKCQkJCQkJImZhaWxfaWZfdW5rbm93biI6IGZhbHNlCgkJCQkJfSwKCQkJCQl7
CgkJCQkJSJpZCI6ICJjcmVkUHJvdGVjdCIsCgkJCQkJSJmYWlsX2lmX3Vua25vd24iOiBmYWxzZQoJ
CQkJCX0KCQkJCV0sCgkJCQkiYXV0aGVudGljYXRvckdldEluZm8iOiB7CgkJCQkJInZlcnNpb25zIjog
WyJVMkZfVjIiLCAiRklET18yXzAiXSwKCQkJCQkiZXh0ZW5zaW9ucyI6IFsiY3JlZFByb3RlY3QiLCAi
aG1hYy1zZWNyZXQiXSwKCQkJCQkiYWFndWlkIjogIjAxMzJkMTEwYmY0ZTQyMDhhNDAzYWI0ZjVmMTJl
ZmU1IiwKCQkJCQkib3B0aW9ucyI6IHsKCQkJCQJInBsYXQiOiAiZmFsc2UiLAoJCQkJCQkicmsiOiAi
dHJ1ZSIsCgkJCQkJSJjbGllbnRQaW4iOiAidHJ1ZSIsCgkJCQkJSJ1cCI6ICJ0cnVlIiwKCQkJCQkJ
InV2IjogInRydWUiLAoJCQkJCQkidXZUb2tlbiI6ICJmYWxzZSIsCgkJCQkJSjb25maWciOiAiZmFs
c2UiCgkJCQJfSwKCQkJCQkibWF4TXNnU2l6ZSI6IDEyMDAsCgkJCQJInBpblV2QXV0aFByb3RvY29s
cyI6IFsxXSwKCQkJCQkibWF4Q3JlZGVudGlhbENvdW50SW5MaXN0IjogMTYsCgkJCQJIm1heENyZWRl
bnRpYWxZExlbmd0aCI6IDEyOCwKCQkJCQkidHJhbnNwb3J0cyI6IFsidXNiIiwgIm5mYyJdLAoJCQkJ
CSJhbGdvcml0aG1zIjogW3sKCQkJCQJSJ0eXBlIjogInB1YmxpYy1rZXkiLAoJCQkJCQJImFsZyI6
IC03CgkJCQJCX0sCgkJCQJCXsKCQkJCQJCSJ0eXBlIjogInB1YmxpYy1rZXkiLAoJCQkJCQJImFs
ZyI6IC0yNTcKCQkJCQJfQoJCQkJCV0sCgkJCQJIm1heEF1dGhlbnRpY2F0b3JDb25maWdMZW5ndGgi
OiAxMDI0LAoJCQkJSjkZWZhdWx0Q3JlZFByb3RlY3QiOiAyLAoJCQkJSjmaXJtd2FyZVZlcnNpb24i
OiA1CgkJCQl9CgkJCX0sCgkJCSJzdGF0dXNSZXBvcnRzIjogW3sKCQkJCQkic3RhdHVzIjogIkZJRE9f
Q0VSVElGSUVEIiwKCQkJCQkiZWZmZWN0aXZlRGF0ZSI6ICIyMDE5LTAxLTA0IgoJCQkJfSwKCQkJCXsK
CQkJCQkic3RhdHVzIjogIkZJRE9fQ0VSVElGSUVEX0wxIiwKCQkJCQkiZWZmZWN0aXZlRGF0ZSI6ICIy
MDIwLTExLTE5IiwKCQkJCQkiY2VydGlmaWNhdGlvbkRlc2NyaXB0b3IiOiAiRklETyBBbGxpYW5jZSBT
YW1wbGUgRklETzIgQXV0aGVudGljYXRvciIsCgkJCQJImNlcnRpZmljYXRlTnVtVmVyIjogIkZJRE8y
MTAwMDIwMTUxMjIxMDAxIiwKCQkJCQkiY2VydGlmaWNhdGlvblBvbGljeVZlcnNpb24iOiAiMS4wLjEi
LAoJCQkJSjZXJ0aWZpY2F0aW9uUmVxdWlyZW1lbnRzVmVyc2lvbiI6ICIxLjAuMSIKCQkJCX0KCQkJ
XSwKCQkJInRpbWVPZkxhc3RTdGF0dXNDaGFuZ2UiOiAiMjAxOS0wMS0wNCIKCQl9CgldCn0
```

```
EXAMPLE: JWT HEADER
{
  "alg": "ES256",
  "typ": "JWT",
  "x5c": [
    "MIICZTCCAgugAwIBAgIBATAKBggqhkjOPQQDAjCBozEnMCUGA1UEAwweRVhBTVBM
    RSBNRFMzIFRFU1QgSU5URVJNRURJQVRFMSIwIAYJKoZIhvcNAQkBFhNleGFtcGxl
    QGV4YW1wbGUuY29tMRQwEgYDVQQKDAtFeGFtcGxlIE9SRzEQMA4GA1UECwwHRXhh
    bXBsZTELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk1ZMRIwEAYDVQQHDAlXYWtlZmll
    bGQwHhcNMjEwNDE5MTEzNTA3WhcNMzEwNDE3MTEzNTA3WjCBpTEpMCcGA1UEAwwg
    RVhBTVBMRSBNRFMzIFNJR05JTkcgQ0VSVElGSUNBVEUxIjAgBgkqhkiG9w0BCQEW
    E2V4YW1wbGVAZXhhbXBsZS5jb20xFDASBgNVBAoMC0V4YW1wbGUgT1JHMRAwDgYD
    VQQLDAdFeGFtcGxlMQswCQYDVQQGEwJVUzELMAkGA1UECAwCTVkxEjAQBgNVBAcM
    CVdha2VmaWVsZDBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABNQJs6wTqixc+S+V
    DAajFlPNat10KEWJE5jcWOvm6qpO9SDAAMZvb4HHrvs+P5YRpHrSlUPdvK+uEQbd
    Wg31P9ujLDAqMAkGA1UdEwQCMAAwHQYDVR0OBBYEFLqsapcXV4ZoVHAnRpPZwQe7
    Yy20MAoGCCqGSM49BAMCA0gAMEUCIQC67za8EIuyRiKgNDXIP1s1aLr3jzH9WVXf
    Hx4bJ+zCsgIgG/tVButOJUU+vvoHIo/otAUAcH5bNHP3uIziDS+PTUc=",
    "MIIEHzCCAgegAwIBAgIBAjANBgkqhkiG9w0BAQsFADCBmzEfMB0GA1UEAwwWRVhB
    TVBMRSBNRFMzIFRFU1QgUk9PVDEiMCAGCSqGSIb3DQEJARYTZXhhbXBsZUBleGFt
    cGxlLmNvbTEUMBIGA1UECgwLRXhhbXBsZSBPUkcxEDAOBgNVBAsMB0V4YW1wbGUx
    CzAJBgNVBAYTAlVTMQswCQYDVQQIDAJNWTESMBAGA1UEBwwJV2FrZWZpZWxkMB4X
    DTIxMDQxOTExMzUwN1oXDTQ4MDkwNDExMzUwN1owgaMxJzAlBgNVBAMMHkVYQU1Q
    TEUgTURTMyBURVNUIElOVEVSTUVESUFURTEiMCAGCSqGSIb3DQEJARYTZXhhbXBs
    ZUBleGFtcGxlLmNvbTEUMBIGA1UECgwLRXhhbXBsZSBPUkcxEDAOBgNVBAsMB0V4
    YW1wbGUxCzAJBgNVBAYTAlVTMQswCQYDVQQIDAJNWTESMBAGA1UEBwwJV2FrZWZp
    ZWxkMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAENGumBbYnFQnTjP1RSfc70hsh
    gbiI1ZtpwQ5n6xRLA/Wq0PSCfLl5qQ+r7dlcK1d3r3vLa+vm6G6vKHGCPEeUzqMv
    MC0wDAYDVR0TBAUwAwEB/zAdBgNVHQ4EFgQUNk6F4RJnGGVFe+0/cbZwfrZd7ZUw
    DQYJKoZIhvcNAQELBQADggIBACnp1fm0FKlWmUtTplLuYg7mps4xP/COu8dnb38u
    1nMDVuOT4+CZaiM9AGz313GD22hjLGrmPuYn86wGOKI3HOrEpsGdMmfy7tTmKX/e
    M/eS3FEDXZnE82Pn5oFIyBT/f8sGuXyOsFZqWBvVdBIIDldCpD4mxMQZZOZtTrlv
    3WvBQMC/dsicOxe3QKXvWHi6Qb/Rhuaip3rPmwMf+4JpnJO+JMPqAaU1cAH8HVsf
    rLAMoKs148j2+cvbpaWmsT5rIoH/ezVrPaG/MOiIgq79w/efuvSi5AX8J+kDoLSE
    f3d5wOgkJYAqUqcRxXTEEtKIzDM6hzaBQFiAWvTn9IlVWgntQamSXvH+txaTF9iE
    lHxUf5INYFVciCpztSrydeHv/OCNRf7/LVricMSlo8Rh+O3yP9V+2uNf3X8sQJNt
    ufrQNaqq18wiXliTLufSn02/g+mkhIUiNKfTOJpvCjKeCnCFcxQU2/XT3Kh3G8gD
    JwsO6EVRjMUJt4AYKze/hEUCwF55IF2m3jHIoCu8jVfj24CeEX5dnfvSr+SVvN5Q
    B0uZ05M4rmyZXyqBm0zK3fR+iE0/ZpInuwLC7X+W82zXlnMkplI3Q+Jxd7jfQ15S
    YNE2K6rvRIT01w0P9ZqyDF7knGKpRlp7Oqxd37bD/VUbWpQ7gIAfsJNH5KBLowHJ
    FFjW"
  ]
}
```

In order to produce the tbsPayload, we first need the base64url-encoded (without padding) JWT Header:

```
EXAMPLE: ENCODED JWT HEADER
ewogICJhbGciOiAiRVMyNTYiLAogICJ0eXAiOiAiSldUIiwKICAieDVjIjogWwogICAgIk1JSUNaVEND
QWd1Z0F3SUJBZ0lCQVRBS0JnZ3Foa2pPUFFRREFqQ0JvekVuTUNVR0ExVUVBd3dlUlZoQlRWQk1SU0JO
UkZNeklGUkZVMVFnU1U1VVJWSk5SVVJKUVZSRk1TSXdJQVlKS29aSWh2Y05BUWtCRmhObGVHRnRjR3hs
UUdWNFlXMXdiR1V1WTI5dE1SUXdFZ1lEVlFRS0RBdEZlR0Z0Y0d4bElFOVNSekVRTUE0R0ExVUVDd3dI
UlhoaGJYQnNaVEVMTUFrR0ExVUVCaE1DVlZNeEN6QUpCZ05WQkFnTUFrMVpNUkl3RUFZRFZRUUhEQWxY
WVd0bFptbGxiR1F3SGhjTk1qRXdOREU1TVRFek5UQTNXaGNOTXpFd05ERTNNVEV6TlRBM1dqQ0JwVEVw
TUNjR0ExVUVBd3dnUlZoQlRWQk1SU0JOUkZNeklGTkpSMDVKVGtjZ1EwVlNWRWxHU1VOQlZFVXhJakFn
QmdrcWhraUc5dzBCQ1FFV0UyVjRZVzF3YkdWQVpYaGhiWEJzWlM1amIyMHhGREFTQmdOVkJBb01DMFY0
WVcxd2JHVWdUMUpITVJBd0RnWURWUVFMREFkRmVHRnRjR3hsTVFzd0NRWURWUVFHRXdKVlV6RUxNQWtH
QTFVRUNBd0NUVmt4RWpBUUJnTlZCQWNNQ1ZkaGEyVm1hV1ZzWkRCWk1CTUdCeXFHU000OUFnRUdDQ3FH
U000OUF3RUhBMElBQk5RSnM2d1RxaXhjK1MrVkRBYWpGbFBOYXQxMEtFV0pFNWpjV092bTZxcE85U0RB
QU1admI0SEhydnMrUDVZUnBIclNsVVBkdksrdUVRYmRXZzMxUDl1akxEQXFNQWtHQTFVZEV3UUNNQUF3
SFFZRFZSME9CQllFRkxxc2FwY1hWNFpvVkhBblJwUFp3UWU3WXkyME1Bb0dDQ3FHU000OUJBTUNBMGdB
TUVVQ0lRQzY3emE4RUl1eVJpS2dORhJUDFzMWFMcjNqekg5V1ZYZkh4NGJKK3pDc2dJZ0cvdFZCdXRP
SlVVK3Z2b0hJby9vdEFVQWNINWJOSFAzdUl6aURTK1BUVWM9IiwKICAgICJNSUlFSHpDQ0FnZWdBd0lC
QWdJQkFqQU5CZ2txaGtpRzl3MEJBUXNGQURDQm16RWZNQjBHQTFVRUF3d1dSVmhCVFZCTVJTQk5SRk16
SUZSRlUxUWdVazlQVkRFaU1DQUdDU3FHU0liM0RRRUpBUllUWlhoaGJYQnNaVUJsZUdGdGNHeGxMbU52
YlRFVU1CSUdBMVVFQ2d3TFJYaGhiWEJzWlNCUFVrY3hFREFPQmdOVkJBc01CMFY0WVcxd2JHVXhDekFK
QmdOVkJBWVRBbFZUTVFzd0NRWURWUVFJREFKTldURVNNQkFHQTFVRUJ3d0pWMkZyWldacFpXeGtNQjRY
RFRJeE1EUXhPVEV4TXpVd04xb1hEVFE0TURrd05ERXhNelV3TjFvd2dhTXhKekFsQmdOVkJBTU1Ia1ZZ
UVUxUVRFVWdUVVJUTXlCVVJWTlVJRWxPVkVWU1RVVkVTVUZVUlRFaU1DQUdDU3FHU0liM0RRRUpBUllU
WlhoaGJYQnNaVUJsZUdGdGNHeGxMbU52YlRFVU1CSUdBMVVFQ2d3TFJYaGhiWEJzWlNCUFVrY3hFREFP
QmdOVkJBc01CMFY0WVcxd2JHVXhDekFKQmdOVkJBWVRBbFZUTVFzd0NRWURWUVFJREFKTldURVNNQkFH
QTFVRUJ3d0pWMkZyWldacFpXeGtNRmt3RXdZSEtvWkl6ajBDQVFZSUtvWkl6ajBEQVFjRFFnQUVOR3Vt
QmJZbkZRblRqUDFSU2ZjNzBoc2hnYmlJMVp0cHdRNW42eFJMQS9XcTBQU0NmTGw1cVErcjdkbGNLMWQz
cjN2TGErdm02RzZ2S0hHQ1BFZVV6cU12TUMwd0RBWURWUjBUQkFVd0F3RUIvekFkQmdOVkhRNEVGZ1FV
Tms2RjRSSm5HR1ZGZSswL2NiWndmclpkN1pVd0RRWUpLb1pJaHZjTkFRUxCUUFEZ2dJQkFDbnAxZm0w
RktsV21VdFRwbEx1WWc3bXBzNHhQL0NPdThkbmIzOHUxbk1EVnVPVDQrQ1phaU05QUd6MzEzR0QyMmhq
TEdybVB1WW44NndHT0tJM0hPckVwc0dkTW1meTd0VG1LWC9lTS9lUzNGRURYWm5FODJQbjVvRkl5QlQv
ZjhzR3VYeU9zRlpxV0J2VmRCSUlEbGRDcEQ0bXhNUVpaT1p0VHJsdjNXdkJRTUMvZHNpY094ZTNRS1h2
V0hpNlFiL1JodWFpcDNyUG13TWYrNEpwbkpPK0pNUHFBYVUxY0FIOEhWc2ZyTEFNb0tzMTQ4ajIrY3Zi
cGFXbXNUNXJJb0gvZXpWclBhRy9NT2lJZ3E3OXcvZWZ1dlNpNUFYOEora0RvTFNFZjNkNXdPZ2tKWUFx
VXFjUnhYVEVFdEtJekRNNmh6YUJRRmlBV3ZUbjlJbFZXZ250UWFtU1h2SCt0eGFURjlpRWxIeFVmNUlO
WUZWY2lDcHp0U3J5ZGVIdi9PQ05SZjcvTFZyaWNNU2xvOFJoK08zeVA5VisydU5mM1g4c1FKTnR1ZnJR
TmFxcTE4d2lYbGlUTHVmU24wMi9nK21raElVaU5LZlRPSnB2Q2pLZUNuQ0ZjeFFVMi9YVDNLaDNHOGdE
SndzTzZFVlJqTVVKdDRBWUt6ZS9oRVVDd0Y1NUlGMm0zakhJb0N1OGpWZmoyNENlRVg1ZG5mdlNyK1NW
dk41UUIwdVowNU00cm15Wlh5cUJtMHpLM2ZSK2lFMC9acEludXdMQzdYK1c4MnpYbG5Na3BsSTNRK0p4
ZDdqZlExNVNZTkUySzZydlJJVDAxdzBQOVpxeURGN2tuR0twUmxwN09xeGQzN2JEL1ZVYldwUTdnSUFm
c0pOSDVLQkxvd0hKRkZqVyIKICBdCn0
```

then we have to append a period (".") and the base64url encoding of the `EncodedMetadataBLOBPayload` (taken from the example in section Metadata BLOB Format):

```
EXAMPLE: TBSPAYLOAD
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsIng1YyI6WyJNSUlDWlRDQ0FndWdBd0lCQWdJQkFUQUtC
Z2dxaGtqT1BRUURBakNCb3pFbk1DVUdBMVVFQXd3ZVJWaEJUVkJNUlNCTlJGTXpJRlJGVTFRZ1NVNVVS
VkpOUlVSSlFWUkZNU0l3SUFZSktvWklodmNOQVFrQkZoTmxlR0Z0Y0d4bFFHVjRZVzF3YkdVdVkyOXRN
UlF3RWdZRFZRUUtEQXRGZUdGdGNHeGxJRTlTUnpFUU1BNEdBMVVFQ3d3SFJYaGhiWEJzWlRFTE1Ba0dB
MVVFQmhNQ1ZWTXhDekFKQmdOVkJBZ01BazFaTVJJd0VBWURWUVFIREFsWFlXdGxabWxsYkdRd0hoY05N
akV3TkRFNU1URXpOVEEzV2hjTk16RXdOREUzTVRFek5UQTNXakNCcFRFcE1DY0dBMVVFQXd3Z1JWaEJU
VkJNUlNCTlJGTXpJRk5KUjA1SlRrY2dRMFZTVkVsR1NVTkJWRVV4SWpBZ0Jna3Foa2lHOXcwQkNRRVdF
MlY0WVcxd2JHVkFaWGhoYlhCc1pTNWpiMjB4RkRBU0JnTlZCQW9NQzBWNFlXMXdiR1VnVDFKSE1SQXdE
Z1lEVlFRTERBZEZlR0Z0Y0d4bE1Rc3dDUVlEVlFRR0V3SlZVekVMTUFrR0ExVUVDQXdDVFZreEVqQVFC
Z05WQkFjTUNWZGhhMlZtYVdWc1pEQlpNQk1HQnlxR1NNNDlBZ0VHQ0NxR1NNNDlBd0VIQTBJQUJOUUpz
NndUcWl4YytTK1ZEQWFqRmxQTmF0MTBLRVdKRTVqY1dPdm02cXBPOVNEQUFNWnZiNEhIcnZzK1A1WVJw
SHJTbFVQZHZLK3VFUWJkV2czMVA5dWpMREFxTUFrR0ExVWRFd1FDTUFBd0hRWURWUjBPQkJZRUZMcXNh
cGNYVjRab1ZIQW5ScFBad1FlN1l5MjBNQW9HQ0NxR1NNNDlCQU1DQTBnQU1FVUNJUUM2N3phOEVJdXlS
aUtnTkRYSVAxczFhTHIzanpIOVdWWGZIeDRiSit6Q3NnSWdHL3RWQnV0T0pVVSt2dm9ISW8vb3RBVUFj
SDViTkhQM3VJemlEUytQVFVjPSIsIk1JSUVIekNDQWdlZ0F3SUJBZ0lCQWpBTkJna3Foa2lHOXcwQkFR
c0ZBRENCbXpFZk1CMEdBMVVFQXd3V1JWaEJUVkJNUlNCTlJGTXpJRlJGVTFRZ1VrOVBWREVpTUNBR0NT
cUdTSWIzREFFSkFSWVRaWGhoYlhCc1pVQmxlR0Z0Y0d4bFxtTnZiVEVVTUlJR0ExVUVDZ3dMUlhoaGJY
```

cUdTSWIzRFFFSkFSWVRaWGhoYlhCc1pVQmxlR0Z0Y0d4bExtTnZiVEVVTUJJR0ExVUVDZ3dMUlhoaGJY
QnNaU0JQVWtjeEVEQU9CZ05WQkFzTUIwVjRZVzF3YkdVeEN6QUpCZ05WQkFZVEFsVlRNUXN3Q1FZRFZR
UUlEQUpOV1RFU01CQUdBMVVFQnd3SlYyRnJaV1pwWld4a01CNFhEVEl4TURReE9URXhNelV3TjFvWERU
UTRNRGt3TkRFeE16VXdOMW93Z2FNeEp6QWxCZ05WQkFNTUhrVllRVTFRVEVVVZ1RVUlRNeUJVUlZOVUlF
bE9WRVZTTVFVWRVNVRlVSVEVpTUNBR0NTcUdTSWIzRFFFSkFSWVRaWGhoYlhCc1pVQmxlR0Z0Y0d4bExt
TnZiVEVVVTUJJR0ExVUVDZ3dMUlhoaGJYQnNaU0JQVWtjeEVEQU9CZ05WQkFzTUIwVjRZVzF3YkdVeEN6
QUpCZ05WQkFZVEFsVlRNUXN3Q1FZRFZRUUlEQUpOV1RFU01CQUdBMVVFQnd3SlYyRnJaV1pwWld4a01G
a3dFd1lIS29aSXpqMENBUVlJS29aSXpqMERBUWNEUWdBRU5HdW1CYlluRlFuVGpQMVJTZmM3MGhzaGdi
aUkxWnRwd1E1bjZ4UkxBL1dxMFBTQ2ZMbDVxUStyN2RsY0sxZDNyM3ZMYSt2bTZHNnZLSEdDUEVlVXpx
TXZNQzB3REFZRFZSMFRCQVV3QXdFQi96QWRCZ05WSFE0RUZnUVVOazZGNFJKbkdHVkZlKzAvY2Jad2Zy
WmQ3WlV3RFFZSktvWklodmNOQVFFTEJRQURnZ0lCQUNucDFmbTBGS2xXbVV0VHBsTHVZZzdtcHM0eFAv
Q091OGRuYjM4dTFuTURWdU9UNCtDWmFpTTlBR3ozMTNHRDIyaGpMR3JtUHVZbjg2d0dPS0kzSE9yRXBz
R2RNbWZ5N3RUbUtYL2VNL2VTM0ZFRFhabkU4MlBuNW9GSXlCVC9mOHNHdVh5T3NGWnFXQnZWZEJJSURs
ZENwRDRteE1RWlpPWnRUcmx2M1d2QlFNQy9kc2ljT3hlM1FLWHZXSGk2UWIvUmh1YWlwM3JQbXdNZis0
SnBuSk8rSk1QcUFhVTFjQUg4SFZzZnJMQU1vS3MxNDhqMitjdmJwYVdtc1Q1cklvSC9lelZyUGFHL01P
aUlncTc5dy9lZnV2U2k1QVg4SitrRG9MU0VvM2Q1d09na0pZQXFVcWNSeFhURUV0S0l6RE02aHphQlFG
aUFXdlRuOUlsVldnbnRRYW1TWHZIK3R4YVRGOWlFbEh4VWY1SU5ZRlZjaUNwenRTcnlkZUh2L09DTlJm
Ny9MVnJpY01TbG84UmgrTzN5UDlWKzJ1TmYzWDhzUUpOdHVmclFOYXFxMTh3aVhsaVRMdWZTbjAyL2cr
bWtoSVVpTktmVE9KcHZDaktlQ25DRmN4UVUyL1hUM0toM0c4Z0RKd3NPNkVWUmpNVUp0NEFZS3plL2hF
VUN3RjU1SUYybTNqSElvQ3U4alZmajI0Q2VFWDVkbmZ2U3IrU1Z2TjVRQjB1WjA1TTRybXlaWHlxQm0w
ekszZlIraUUwL1pwSW51d0xDN1grVzgyelhsbk1rcGxJM1ErSnhkN2pmUTE1U1lOORTJLNnJ2UklUMDF3
MFA5WnF5REY3a25HS3BSbHA3T3F4ZDM3YkQvVlViV3BRN2dJQWZzSk5INUtCTG93SEpGRmpXIl19.eyJ
sZWdhbEhlYWRlciI6IlJldHJpZXZhbCBhbmQgdXNlIG9mIHRoaXMgQkxPQiBpbmRpY2F0ZXMgYWNjZXB
0YW5jZSBvZiB0aGUgYXBwcm9wcmlhdGUgYWdyZWVtZW50IGxvY2F0ZWQgYXQgaHR0cHM6Ly9maWRvYWx
saWFuY2Uub3JnL21ldGFkYXRhL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoiMTIzNCM1Njc4IiwibWV0YWRhdGFTdGF
0ZW1lbnQiOnsibGVnYWxIZWFkZXIiOiJodHRwczovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWhlYWRlci8iLCJkZXNjcmlwdGlvbiI6IkZJRE8gQWxsaWFuY2U
gU2FtcGxlIFVBRiBBdXRoZW50aWNhdG9yIiwiYWFpZCI6IjEyMzQjNTY3OCIsImFsdGVybmF0aXZlRGV
zY3JpcHRpb25zIjp7InJ1LVJVIjoi0J_RgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YLQuNGE0LjQutC
w0YLQvtGA0LAg0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGxlIFVBRiBhdXRoZW50aWN
hdG9yIGRlIEZJRE8gQWxsaWFuY2UifSwiYXV0aGVudGljYXRvclZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjoidWFmIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxLCJtaW5vciI6MH0seyJtYWpvciI
6MSwibWlub3IiOjF9XSwiYXV0aGVudGljYXRpb25BbGdvcml0aG1zIjpbInNlY3AyNTZyMV9lY2RzYV9
zaGEyNTZfcmF3Il0sInB1YmxpY0tleUFsZ0FuZEVuY29kaW5ncyI6WyJlY2NfeDk2Ml9yYXciXSwiYXR
0ZXN0YXRpb25UeXBlcyI6WyJiYXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcmlmaWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaW50X2ludGVybmFsIiwiYmFEZXNjIjp7InN
lbGZBdHRlc3RlZEZBUiI6MC4wMDAwMiwibWF4UmV0cmllcyI6NSwiYmxvY2tTbG93ZG93biI6MzAsIm1
heFRlbXBsYXRlcyI6NX19XV0sImtleVByb3RlY3Rpb24iOlsiaGFyZHdhcmUiLCJ0ZWUiXSwiaXNLZXl
SZXN0cmljdGVkIjp0cnVlLCJtYXRjaGVyUHJvdGVjdGlvbiI6WyJ0ZWUiXSwiY3J5cHRvU3RyZW5ndGg
iOjEyOCwiYXR0YWNobWVudEhpbnQiOlsiaW50ZXJuYWwiXSwidGNEaXNwbGF5IjpbImFueSIsInRlZSJ
dLCJ0Y0Rpc3BsYXlDb250ZW50VHlwZSI6ImltYWdlL3BuZyIsInRjRGlzcGxheVBOR0NoYXJhY3Rlcml
zdGljcyI6W3sid2lkdGgiOjMyMCwiaGVpZ2h0Ijo0ODAsImJpdERlcHRoIjoxNiwiY29sb3JUeXBlIjo
yLCJjb21wcmVzc2lvbiI6MCwiZmlsdGVyIjowLCJpbnRlcmxhY2UiOjB9XSwiYXR0ZXN0YXRpb25Sb29
0Q2VydGlmaWNhdGVzIjpbIk1JSUNQVENDQWVPZ0F3SUJBZ0lKQU91ZXh2VTNPeTJ3TUFvR0NDcUdTTTQ
5QkFNQ01Ic3hJREFlQmdOVkJBTU1GMU5oYlhCc1pTQkJkSFJsYzNSaGRHbHZiaUJTYjI5ME1SWXdGQVl
EVlFRS0RBMUdTVVJQSUVGc2JHbGhibU5sTVJFd0R3WURWUVFMREFoVlFVWWdWRmRITERFU01CQUdBMVV
FQnd3SlVHRnNieUJCYkhSdk1Rc3dDUVlEVlFRSURBSkRRVEVMTUFrR0ExVUVCaE1DVlZNd0hoY05NVFF
3TmpFNE1UTXpNek15V2hjTk5ERXhNVEF6TVRNek16TXlXakI3TVNBd0hnWURWUVFEREJkVFlXMXdiR1V
nUVhSMFpYTjBZWFJwYjI0Z1VtOXZkREVXTUJRR0ExVUVDZ3dOUmtsRVR5QkJiR3hwWVc1alpURVJNQTh
HQTFVRUN3d0lWVUZHSUZSWFJ5d3hFakFRQmdOVkJBY01DVkJoYkc4Z1FXeDBiekVMTUFrR0ExVUVDQXd
DUTBFeEN6QUpCZ05WQkFZVEFsVlRNRmt3RXdZSEtvWkl6ajBDQVFZSUtvWkl6ajBEQVFjRFFnQUVIOGh
2MkQwSFhhNTkvQm1wUTdSWmVoTC9GTUd6RmQxUUJnOXZBVXBPWjNham51UTk0UFI3YU16blVTQnI
4ZkhZRHJxT0JiNThweEdxSEpSeVgvNk5RTUUwd0hRWURWUjBPQkJZRUZQb0hBM0NMaHhGYkMwSXQ3ekU
0dzhoazVFSi9NQjhHQTFVZEl3UVlNQmFBRlBvSEEzQ0xoeEZiQzBJdDd6RTR3OGhrNUVKL01Bd0dBMVV
kRXdRRk1BTUJBZjh3Q2dZSUtvWkl6ajBFQXdJRFNBQXdSUUloQUowNlFTWHQ5aWhJYkVLWUtJanNQa3J
pVmRMSWd0ZnNiRFN1N0VySmZ6cjRBaUJxb1lDWmYwK3pJNTVhUWVBSGpJekE5WG02M3JydUF4Qlo5cHM
5ejJYTmxRPT0iXSwiaWNvbiI6ImRhdGE6aW1hZ2UvcG5nO2Jhc2U2NCxpVkJPUncwS0dnb0FBQUFOU1V
oRVVnQUFBRThBQUFBdkNBWUFBQUNpd0pmY0FBQUFBWE5TUjBJQXJzNGM2UUFBQUFSblFVMUJBQUN4and
2OFlRVUFBQUFKY0VoWmN3QUFEc01BQUE3REFjZHZxR1FBQUFhaFNVUkJWR2hEN1pyNWJ4UmxHTWY5S3p
UQjhBTS9ZRWhFMlc3cFFaa1dLS0JjbFNwSEFUbEVVMQVJFN2tORUNDTlRNGa1dLMENLS1NNNDRklzS0JjZ1Z
DRFdHTkVTZEFZaWR3Z2dnSkpUmlNaEZjLzR3eTg4ODR6dTlOZGxuR1RmWkpQQm4zbk8rKzg4OTMzZ
lQkJ4K1BxQ3pKa1RVdkJiTG1wVURXdkJUSW1wY0NTWnYTEJkWDlSMDVTazE5YmI1YXRmNTk5ZkcrL2V

yQTU0MXE0N2FQMUxMVmE5U0l5Vk5VaThJaThkNWtHVHNpMzBORnY3YWk5bjdRWlBNd2JkeXMyZXJVMlh
NcVVkeTgrWmNhTm1HaW1FOHlYTjNSVWQzYTE4bkYwZlVsb3ZaKzBDVHpXcGQyVmorZU9tMWJFeXk2RHg
0aTVwVU1HV3ZlbzUwNnEyMjdkdHVXQkl1ZmZyNm9XcFYwRlBOTGhvdzE3NTFObTIxTHZQSDNyVnRXamZ
6NjZMZnFsOHRYN0ZSbDlZRlNYc21Tc2ViOWNlT0diWWs3TU5VY0dQZzhac2JNZTlyZlFVYWFWL0pNWDl
zcWR6RENTdnAwa1pIbVRaZzl4N2JMSGNNblRoYjE2ZUorbVZmUXE4eWFVWlFORzY0aVhaKzAva3E2dU9
aRk8wUXRhdGRXS2ZYblJROTlCajkxUjVPSUZuazU0ak4wbWtVaXFsTzNYRFcrTWwrOThtS0I2dFc3cld
wWmNQYyswemc0dExyWWxVYzg2RTZlR0RqSU11YlZwY3VzZWFyZmdJWUdSazZicmhaVnIvSmNIem9vTDc
1NTBqZWRMRXhvcFdjQXBpMlpVcWh1N0pMdnJWc1FVODF6a3pPUGVlbU1SWXZWdVFzWDdQYmlEUVk1SnZ
ab25mdEsrMVZZOEg5dXR4NTMwaDBvYitqbVJZcWo2b3VhWXZFZW5XL1dsWWpwOGN3Yk1tNjgydFB3cVc
xUjR0ai8yU0gxM0lSSllsNG1vWnZYcGlTcURyN2RYdFFIeGEvUEszLytCV3NLMWRUZ0h1NlY4dFFKM2J
3Rmt3cEZyVU9RNTBzMXIzbGV2bTh6WmNxMTcrQkJhdzdLOGxFSzVxemtZZWFyazlBOHA3UDNHekRLK25
kM0RRb3crNlVDOFNWTjgyaXV2MzhpbTdOdGFYdFYxQ1ZxNlJndzRwa3NtYmRpM2J1MkRlN1lmYUJCeGN
xZnZxUHJVakZRTlRRMjJsZmRVVlZUNjhyVEpLRjVEblNtVWpnZHFnNG1TUzlwbXNmREpSM0c2VG9IMGl
XOWFWN0xXTEhZWEtsbFREdDBMVEF0a1lJYWFtcDFRalZ2Kyt1eUdVeFZkSjBETlZYU20rYjFxUnhwbDg
0ZGRmWDFMcDFPL2Q2OXRzb2QwdnM1aEdyZTl4dThvK2ZwTFIxY0doTlRENlo1N0M5S01XWGVmSmRPWjk
0YmI5b3FkMVJPblM3cUlUVHpIaW1NcWl2Yk8zZzBEZFZ5azNXUUJoQnp0SzM1WUtOZE9uYzhPM2FjUzZ
mRFpGZ0thWExzRUpwNXJkcmxpQnFwODljSmNzL203VHZzMHJrakdmTjRiMGtQb1puM1VKdUlPcm5aMjJ
5UDFmbXZVeCtPNWdTcWViVjFtK3pTdVlOVmhxN1RXYkRpTFZ2bGpwbExsb3A2Q0xYUCsycXR2R0xJTC8
xdmltSVNkTUJnelNvRlp5dTZUcWQranp4Z3NQYVY5QkNxZWUvTmpZazZ2NmxLOWN3aVVjL1NUdGYxSER
wTTNiNTkyeTdoM1RoeDVveks2OUhMcFlXdUF3YXFTNWN2MjZxN2NlYjhlZlZZYVJlUDNpRlU4emoxa25
Td1pYSE1tbkNqWTBPZ2FsbzdVUWZTQ00zcVFRcjJIL1hGUDdzc1h4NDVZbDkxQnllQ2VwNG1vWm9IKzF
mRzN4RDR0VDd4OGt3eWo4bndiOWV2MjZWMEI2ZCs3SDR6S3Z1ZEFINTM3RmpxeXpPSGRKbkhFdXptWHE
vV2p4T2J2Tk1idjduaHl3c1gyYVZzV3RDOCs0OGFMZWFwRTdwNXdLWmkwQTJBUVJWNW52UjRFK3VKYyt
iNjFrQXBxSW54QmdtZC80VjVRUC9tdDE4SERDN3NSSGZ0bWV1NWxtaFYwcm4vQUxYMjMyYnFkNEJGbkR
4N1ZpMWNXUzJ1ZmYwSWJCNDdxZXh4bVVqOVF1dFlqdXBkM3RZRDZhYldCQk1yaCthcE5iT0tyTkYxK3V
nQ2E0cmlYR2Z3TVBQdFZpYXZoVTNZTU9BQW51VWIvUjA3TDB5T1NlT2FkRTg4QXBzWEZHZmYzMHluaGx
KZ001MUNVNnZOOUV6Z25wdkhCRlV5aVZyYWVQaXdKNTNERjVaVFpub21FTmc4NWtOVWQyb0ppMldwcjR
PbW1rZk40eDR6SGZpVkZjOER2OE56dWhOcU9pZGlsR3ZBNkRHdWVad083OEFBUW42Y2lFazYrcnc1VmN
2anZxTkRZUE9vSVV3YUtTaHJ4QXVYTGxrSDRhWXVHZk1ZRGMxMFdGNVRhMzFoUEpPZmNVaHJVL0psSU5
pNmM2ZWxSWWRCcG82KytZZmp4NjFsR05mUm00TUQ1ckoxajNGb0dIbmpEU0JOYXJZVWdNTHlNc3pLcGI
3dFhwb0hmUHM4aDNXcDFMek5mTms1NFh4QzF3REdVbVl6WFllZmg2ei9jS3RWbTRFQnhhOVZRR0R6WXI
zTHJVTVJqSEVLa2s3emFGS1lRQTJoR1FVMXorODVORldwWERya3ozdngxMEdxeFE2QnplTmJvQms1bjh
rNG5lYlJoK2sxaFdmeFRGMEQxRXlXVXM1bnYrZGdRcUtheHp1Q2RFMGlzSGwwMk5ROGFoMG1YcjEyTGE
zbTBmOXdpazkrd0xOVE1ZLzg2TVBvOHlpMzFPZnhtVDZQV29xRzkrRFp1a1luYTU2bVNadDVXV1N5NXF
WQTFyd1V5SnFYQWxuemtpYWkvZ0hTRDdSa1R5aWhvZ0FBQUFCSlJVNUVya0pnZ2c9PSJ9LCJzdGF0dXN
SZXBvcnRzIjpbeyJzdGF0dXMiOiJGSURPX0NFUlRJRklFRCIsImVmZmVjdGl2ZURhdGUiOiIyMDE0LTA
xLTA0In1dLCJ0aW1lT2ZMYXN0U3RhdHVzQ2hhbmdlIjoiMjAxNC0wMS0wNCJ9LHsiYWFndWlkIjoiMDE
zMmQxMTAtYmY0ZS00MjA4LWE0MDMtYWI0ZjVmMTJlZmU1IiwibWV0YWRhdGFTdGF0ZW1lbnQiOnsibGV
nYWxIZWFkZXIiOiJodHRwczovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV0YWRhdGEtc3RhdGV
tZW50LWxlZ2FsLWhlYWRlci8iLCJkZXNjcmlwdGlvbiI6IkZJRE8gQWxsaWFuY2UgU2FtcGxlIEZJRE8
yIEF1dGhlbnRpY2F0b3IiLCJhYWd1aWQiOiIwMTMyZDExMC1iZjRlLTQyMDgtYTQwMy1hYjRmNWYxMmV
mZTUiLCJhbHRlcm5hdGl2ZURlc2NyaXB0aW9ucyI6eyJydS1SVSI6ItCf0YDQuNC80LXRgCBGSURPMiD
QsNGD0YLQtdC90YLQuNGE0LjQutCw0YLQvtGA0LAg0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJ
FeGVtcGxlIEZJRE8yIGF1dGhlbnRpY2F0b3IgZGUgRklETyBBbGxpYW5jZSIsInpoLUNOIjoi5L6G6Ie
qRklETyBBbGxpYW5jZeeahOekuuS-i0ZJRE8y6Lqr5Lu96amX6K2J5ZmoIn0sInByb3RvY29sRmFtaWx
5IjoiZmlkbzIiLCJzY2hlbWEiOjMsImF1dGhlbnRpY2F0b3JWZXJzaW9uIjo1LCJ1cHYiOlt7Im1ham9
yIjoxLCJtaW5vciI6MH1dLCJhdXRoZW50aWNhdGlvbkFsZ29yaXRobXMiOlsic2VjcDI1NnIxX2VjZHN
hX3NoYTI1Nl9yYXciLCJyc2Fzc2FfcGtjc3YxNV9zaGEyNTZfcmF3Il0sInB1YmxpY0tleUFsZ0FuZEV
uY29kaW5ncyI6WyJjb3NlIl0sImF0dGVzdGF0aW9uVHlwZXMiOlsiYmFzaWNfZnVsbCJdLCJ1c2VyVmV
yaWZpY2F0aW9uRGV0YWlscyI6W1t7InVzZXJWZXJpZmljYXRpb25NZXRob2QiOiJub25lIn1dLFt7InV
zZXJWZXJpZmljYXRpb25NZXRob2QiOiJwcmVzZW5jZV9pbnRlcm5hbCJ9XSxbeyJ1c2VyVmVyaWZpY2F
0aW9uTWV0aG9kIjoicGFzc2NvZGVfZXh0ZXJuYWwiLCJjYURlc2MiOnsiYmFzZSI6MTAsIm1pbkxlbmd
0aCI6NH19XSxbeyJ1c2VyVmVyaWZpY2F0aW9uTWV0aG9kIjoicGFzc2NvZGVfZXh0ZXJuYWwiLCJjYUR
lc2MiOnsiYmFzZSI6MTAsIm1pbkxlbmd0aCI6NH19LHsidXNlclZlcmlmaWNhdGlvbk1ldGhvZCI6InB
yZXNlbmNlX2ludGVybmFsIn1dXSwia2V5UHJvdGVjdGlvbiI6WyJoYXJkd2FyZSIsInNlY3VyZV9lbGV
tZW50Il0sIm1hdGNoZXJQcm90ZWN0aW9uIjpbIm9uX2NoaXAiXSwiY3J5cHRvU3RyZW5ndGgiOjEyOCw
iYXR0YWNobWVudEhpbnQiOlsiZXh0ZXJuYWwiLCJ3aXJlZCIsIndpcmVsZXNzIiwibmZjIl0sInRjRGl
zcGxheSI6W10sImF0dGVzdGF0aW9uUm9vdENlcnRpZmljYXRlcyI6WyJNSUlDUFRDQ0FlT2dBd0lCQWd
JSkFPdWV4dlUzT3kyd01Bb0dDQ3FHU000OUJBTUNNSHN4SURBZUJnTlZCQU1NRjFOaGJYQnNaU0JCZEh
SbGMzUmhkR2x2YmlCU2IyOTBNUll3RkFZRFZRUUtEQTFHU1VSUElFRnNiR2xoYm1ObE1SRXdEd1lEVlF
RTERBaFZRVVlnVkZkSExERVNNQkFHQTFVRUJ3d0pVR0ZzYnlCQmJIUnZNUXN3Q1FZRFZRUUlEQUpEUVR
FTE1Ba0dBMVVFQmhNQ1ZWTXdIaGNOTVRRd05qRTRNVE16TXpNeVdoY05OREV4TVRBek1UTXpNek15V2p

CN01TQXdIZ1lEVlFRRERCZFRZVzF3YkdVZ1FYUjBaWE4wWVhScGIyNGdVbTl2ZERFV01CUUdBMVVFQ2d
3TlJrbEVUeUJCYkd4cFlXNWpaVEVSTUE4R0ExVUVDd3dJVlVGR0lGUlhSeXd4RWpBUUJnTlZCQWNNQ1Z
CaGJHOGdRV3gwYnpFTE1Ba0dBMVVFQ0F3Q1EwRXhDekFKQmdOVkJBWVRBbFZUTUZrd0V3WUhLb1pJemo
wQ0FRWUlLb1pJemowREFRY0RRZ0FFSDhodjJEMEhYYTU5L0JtcFE3UlplaEwvRk1HekZkMVFCZzl2QVV
wT1ozYWpudVE5NFBSN2FNekgzM25VU0JyOGZIWURycU9CYjU4cHhHcUhKUnlYLzZOUU1FNHdIUVlEVlI
wT0JCWUVGUG9IQTNDTGh4RmJDMEl0N3pFNHc4aGs1RUovTUI4R0ExVWRJd1FZTUJhQUZQb0hBM0NMaHh
GYkMwSXQ3ekU0dzhoazVFSi9NQXdHQTFVZEV3UUZNQU1CQWY4d0NnWUlLb1pJemowRUF3SURSQUF3UlF
JaEFKMDZRU1h0OWloSWJFS1lLSWpzUGtyaVZkTElndGZzYkRTdTdFckpmenI0QWlCcW9ZQ1pmMCt6STU
1YVFlQUhqSXpBOVhtNjNycnVBeEJaOXBzOXoyWE5sUT09Il0sImljb24iOiJkYXRhOmltYWdlL3BuZzt
iYXNlNjQsaVZCT1J3MEtHZ29BQUFBTlNVaEVVZ0FBQUU4QUFBQXZDQVlBQUFDaXdKZmNBQUFBQVhOU1I
wSUFyczRjNlFBQUFBUm5RVTFCQUFDeGp3djhZUVVBQUFBSmNFaFpjd0FBRHNNQUFBN0RBY2R2cUdRQUF
BYWhTVVJCVkdoRDdacjVieFJsR01mOUt6VEI4QU0vWUVoRTJXN3BRWmNXS0tDY2x2TcEhBVGxFTEFFSRTd
rTkVDQ0EzRmtXSzBSK2NBMG1TMzYvdTNVEVWS0dCMjlFUjBuV1dnTDI2NEJEaEZQbDg3ajZlVwI2Rmg1VkwrCZVmZ0xYeHd6ZUdsWVAyZHdDQzBnbnl0cnVsZzkvNEdQNzZscmdDQVB3cGJYTG9YdDlyQzc0UnZmSGs1N0lEZDJRYmZPU1Q3SHdtMHNjVzU4UWJlRHdsNmowcVBBQ0lyM2dEa29QckYzZVhaQzlOdXgyWT0=

```
MMSIsImVmZmVjdGl2ZUKndGU1U1IyMDIwLTExLTE5T1wiY2VydGlmaWNhdGlvbkRlc2NyaXB0b3I1U1J
GSURPIEFsbGlhbmNlIFNhbXBsZSBGSURPMiBBdXRoZW50aWNhdG9yIiwiY2VydGlmaWNhdGVOdW1iZXI
iOiJGSURPMjEwMDAyMDE1MTIyMTAwMSIsImNlcnRpZmljYXRpb25Qb2xpY3lWZXJzaW9uIjoiMS4wLjE
iLCJjZXJ0aWZpY2F0aW9uUmVxdWlyZW1lbnRzVmVyc2lvbiI6IjEuMC4xIn1dLCJ0aW1lT2ZMYXN0U3R
hdHVzQ2hhbmdlIjoiMjAxOS0wMS0wNCJ9XX0
```

and finally we have to append another period (".") followed by the base64url-encoded signature.

```
EXAMPLE: JWT
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsIng1YyI6WyJNSUlDWlRDQ0FndWdBd0lCQWdJQkFUQUtC
Z2dxaGtqT1BRUURBakNCb3pFbk1DVUdBMVVFQXd3ZVJWaEJUVkJNUlNCTlJGTXpJRlJGVTFRZ1NVNVVS
VkpOUlVSSlFWUkZNU0l3SUFZSktvWklodmNOQVFrQkZoTmxlR0Z0Y0d4bFFHVjRZVzF3YkdVdVkyOXRN
UlF3RWdZRFZRUUtEQXRGZUdGdGNHeGxJRTlTUnpFUU1BNEdBMVVFQ3d3SFJYaGhiWEJzWlRFTE1Ba0dB
MVVFQmhNQ1ZWTXhDekFKQmdOVkJBZ01BazFaTVJJd0VBWURWUVFIREFsWFlXdGxabWxsYkdRd0hoY05N
akV3TkRFNU1URXpOVEEzV2hjTk16RXdOREUzTVRFek5UQTNXakNCcFRFcE1DY0dBMVVFQXd3Z1JWaEJU
VkJNUlNCTlJGTXpJRk5KUjA1SlRrY2dRMFZTVkVsR1NVTkJWRVV4SWpBZ0Jna3Foa2lHOXcwQkNRRVdF
MlY0WVcxd2JHVkFaWGhoYlhCc1pTNWpiMjB4RkRBU0JnTlZCQW9NQzBWNFlXMXdiR1VnVDFKSE1SQXdE
Z1lEVlFRTERBZEZlR0Z0Y0d4bE1Rc3dDUVlEVlFRR0V3SlZVekVMTUFrR0ExVUVDQXdDVFZreEVqQVFC
Z05WQkFjTUNWZGhhMlZtYVdWc1pEQlpNQk1HQnlxR1NNNDlBZ0VHQ0NxR1NNNDlBd0VIQTBJQUJOUUUz
NndUcWl4YytTK1ZEQWFqRmxQTmF0MTBLRVdKRTVqY1dPdm02cXBPOVNEQUFNWnZiNEhIcnZzK1A1WVJw
SHJTbFVQZHZLK3VFUWJkV2czMVA5dWpMREFxTUFrR0ExVWRFd1FDTUFBd0hRWURWUjBPQkJZRUZMcXNh
cGNYVjRab1ZIQW5ScFBad1FlN1l5MjBNQW9HQ0NxR1NNNDlCQU1DQTBnQU1FVUNJUUM2N3phOEVJdXlS
aUtnTkRYSVAxczFhTHIzanpIOVdWWGZIeDRiSit6Q3NnSWdHL3RWQnV0T0pVVSt2dm9ISW8vb3RBVUFj
SDViTkhQM3VJemlEUytQVFVjPSIsIk1JSUVIekNDQWdlZ0F3SUJBZ0lCQWpBTkJna3Foa2lHOXcwQkFR
c0ZBRENCbXpFZk1CMEdBMVVFQXd3V1JWaEJUVkJNUlNCTlJGTXpJRlJGVTFRZ1VrOVBWREVpTUNBR0NT
cUdTSWIzRFFFSkFSWVRaWGhoYlhCc1pVQmxlR0Z0Y0d4bExtTnZiVEVVTUJJR0ExVUVDZ3dMUlhoaGJY
QnNaU0JQVWtjeEVEQU9CZ05WQkFzTUIwVjRZVzF3YkdVeEN6QUpCZ05WQkFZVEFsVlRNUXN3Q1FZRFZR
UUlEQUpOV1RFU01CQUdBMVVFQnd3SlYyRnJaV1pwWld4a01CNFhEVEl4TURReE9URXhNelV3TjFvWERU
UTRNRGt3TkRFeE16VXdOMW93Z2FNeEp6QWxCZ05WQkFNTUhrVllRVTFRVEVVZ1RVUlRNeUJVUlZOVUlF
bE9WRVZTVFVWRVNVRlVSVEVpTUNBR0NTcUdTSWIzRFFFSkFSWVRaWGhoYlhCc1pVQmxlR0Z0Y0d4bExt
TnZiVEVVTUJJR0ExVUVDZ3dMUlhoaGJYQnNaU0JQVWtjeEVEQU9CZ05WQkFzTUIwVjRZVzF3YkdVeEN6
QUpCZ05WQkFZVEFsVlRNUXN3Q1FZRFZRUUlEQUpOV1RFU01CQUdBMVVFQnd3SlYyRnJaV1pwWld4a01G
a3dFd1lIS29aSXpqMENBUVlJS29aSXpqMERBUWNEUWdBRU5HdW1CYlluRlFuVGpQMVJTZmM3MGhzaGdi
aUkxWnRwd1E1bjZ4UkxBL1dxMFBTQ2ZMbDVxUStyN2RsY0sxZDNyM3ZMYSt2bTZHNnZLSEdDUEVlVXpx
TXZNQzB3REFZRFZSMFRCQVV3QXdFQi96QWRCZ05WSFE0RUZnUVVOazZGNFJKbkdHVkZlKzAvY2Jad2Zy
WmQ3WlV3RFFZSktvWklodmNOQVFFTEJRQURnZ0lCQUNucDFmbTBGS2xXbVV0VHBsTHVZZzdtcHM0eFAv
Q091OGRuYjM4dTFuTURWdU9UNCtDWmFpTTlBR3ozMTNHRDIyaGpMR3JtUHVZbjg2d0dPS0kzSE9yRXBz
R2RNbWZ5N3RUbUtYL2VNL2VTM0ZFRFhabkU4MlBuNW9GSXlCVC9mOHNHdVh5T3NGWnFXQnZWZEJJSURs
ZENwRDRteE1RWlpPWnRUcmx2M1d2QlFNQy9kc2ljT3hlM1FLWHZXSGk2UWIvUmh1YWlwM3JQbXdNZis0
SnBuSk8rSk1QcUFhVTFjQUg4SFZzZnJMQU1vS3MxNDhqMitjdmJwYVdtc1Q1cklvSC9lelZyUGFHL01P
aUlncTc5dy9lZnV2U2k1QVg4SitrRG9MU0VmM2Q1d09na0pZQXFVcWNSeFhURUV0S0l6RE02aHphQlFG
aUFXdlRuOUlsVldnbnRRYW1TWHZIK3R4YVRGOWlFbEh4VWY1SU5ZRlZjaUNwenRTcnlkZUh2L09DTlJm
Ny9MVnJpY01TbG84UmgrTzN5UDlWKzJ1TmYzWDhzUUpOdHVmclFOYXFxMTh3aVhsaVRMdWZTbjAyL2cr
bWtoSVVpTktmVE9KcHZDaktlQ25DRmN4UVUyL1hUM0toM0c4Z0RKd3NPNkVWUmpNVUp0NEFZS3plL2hF
VUN3RjU1SUYybTNqSElvQ3U4alZmajI0Q2VFWDVkbmZ2U3IrU1Z2TjVRQjB1WjA1TTRybXlaWHlxQm0w
ekszZlIraUUwL1pwSW51d0xDN1grVzgyelhsbk1rcGxJM1ErSnhkN2pmUTE1U1lORTJLNnJ2UklUMDF3
MFA5WnF5REY3a25HS3BSbHA3T3F4ZDM3YkQvVlViV3BRN2dJQWZzSk5INUtCTG93SEpGRmpXIl19.eyJ
sZWdhbEhlYWRlciI6IlJldHJpZXZhbCBhbmQgdXNlIG9mIHRoaXMgQkxPQiBpbmRpY2F0ZXMgYWNjZXB
0YW5jZSBvZiB0aGUgYXBwcm9wcmlhdGUgYWdyZWVtZW50IGxvY2F0ZWQgYXQgaHR0cHM6Ly9maWRvYWx
saWFuY2Uub3JnL21ldGFkYXRhL21ldGFkYXRhLWxlZ2FsLXRlcm1zLyIsIm5vIjoxNSwibmV4dFVwZGF
0ZSI6IjIwMjAtMDMtMzAiLCJlbnRyaWVzIjpbeyJhYWlkIjoiMTIzNCM1Njc4IiwibWV0YWRhdGFTdGF
0ZW1lbnQiOnsibGVnYWxIZWFkZXIiOiJodHRwczovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV
0YWRhdGEtc3RhdGVtZW50LWxlZ2FsLWhlYWRlci8iLCJkZXNjcmlwdGlvbiI6IkZJRE8gQWxsaWFuY2U
gU2FtcGxlIFVBRiBBdXRoZW50aWNhdG9yIiwiYWFpZCI6IjEyMzQjNTY3OCIsImFsdGVybmF0aXZlRGV
zY3JpcHRpb25zIjp7InJ1LVJVIjoi0J_RgNC40LzQtdGAIFVBRiDQsNGD0YLQtdC90YLQuNGE0LjQutC
w0YLQvtGA0LAg0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJFeGVtcGxlIFVBRiBhdXRoZW50aWN
hdG9yIGRlIEZJRE8gQWxsaWFuY2UifSwiYXV0aGVudGljYXRvclZlcnNpb24iOjIsInByb3RvY29sRmF
taWx5IjoidWFmIiwic2NoZW1hIjozLCJ1cHYiOlt7Im1ham9yIjoxLCJtaW5vciI6MH0seyJtYWpvciI
6MSwibWlub3IiOjF9XSwiYXV0aGVudGljYXRpb25BbGdvcml0aG1zIjpbInNlY3AyNTZyMV9lY2RzYV9
zaGEyNTZfcmF3Il0sInB1YmxpY0tleUFsZ0FuZEVuY29kaW5ncyI6WyJlY2NfeDk2Ml9yYXciXSwiYXR
0ZXN0YXRpb25UeXBlcyI6WyJiYXNpY19mdWxsIl0sInVzZXJWZXJpZmljYXRpb25EZXRhaWxzIjpbW3s
idXNlclZlcmlmaWNhdGlvbk1ldGhvZCI6ImZpbmdlcnByaW50X2ludGVybmFsIiwiYmFEZXNjIjp7InN
```

```
lbGZBdHRlc3RlZEZBU1i6MC4wMDAwM1w1bWF4UmV0cmllcyI6NSw1YmxvY2tIbG93ZG93bi16MzAsIml
heFRlbXBsYXRlcyI6NX19XV0sImtleVByb3RlY3Rpb24iOlsiaGFyZHdhcmUiLCJ0ZWUiXSwiaXNLZXl
SZXN0cmljdGVkIjp0cnVlLCJtYXRjaGVyUHJvdGVjdGlvbiI6WyJ0ZWUiXSwiY3J5cHRvU3RyZW5ndGg
iOjEyOCwiYXR0YWNobWVudEhpbnQiOlsiaW50ZXJuYWwiXSwidGNEaXNwbGF5IjpbImFueSIsInRlZSJ
dLCJ0Y0Rpc3BsYXlDb250ZW50VHlwZSI6ImltYWdlL3BuZyIsInRjRGlzcGxheV8yY4Z1FXeDBiekVMTUFrR0ExVUVDaXd
zdGljcyI6W3sid2lkdGgiOjMyMCwiaGVpZ2h0Ijo0ODAsImJpdERlcHRoIjoxNiwiY29sb3JUeXBlIjo
yLCJjb21wcmVzc2lvbiI6MCwiZmlsdGVyIjowLCJpbnRlcmxhY2UiOjB9XSwiYXR0ZXN0YXRpb25Sb29
0Q2VydGlmaWNhdGVzIjpbIk1JSUNQVENDQWVPZ0F3SUJBZ0lKQU91ZXh2VTNPeTJ3TUFvR0NDcUdTTTQ
5QkFNQ01Ic3hJREFlQmdOVkJBTU1GMU5oYlhCc1pTQkJkSFJsYzNSaGRHbHZiaUJTYjI5ME1SWXdGQVl
EVlFRS0RBMUdTVVJQSUVGc2JHbGhibU5sTVJFd0R3WURWUVFMREFoVlFVWWdWRmRITERFU01CQUdBMVV
FQnd3SlVHRnNieUJCYkhSdk1Rc3dDUVlEVlFRSURBSkRRVEVMTUFrR0ExVUVCaE1DVlZNd0hoY05NVFF
3TmpFNE1UTXpNek15V2hjTk5ERXhNVEF6TVRNek16TXlXakI3TVNBd0hnWURWUVFEREJkVFlXMXdiR1V
nUVhSMFpYTjBZWFJwYjI0Z1VtOXZkREVXTUJRR0ExVUVDZ3dOUmtsRVR5QkJiR3hwWVc1alpURVJNQTh
HQTFVRUN3d0lWVUZHSUZSWFJ5d3hFakFRQmdOVkJBY01DVkJoYkc4Z1FXeDBiekVMTUFrR0ExVUVDQXd
DUTBFeEN6QUpCZ05WQkFZVEFsVlRNRmt3RXdZSEtvWkl6ajBDQVFZSUtvWkl6ajBEQVFjRFFnQUVIOGh
2MkQwSFhhNTkvQm1wUTdSWmVoTC9GTUd6RmQxUUJnOXZBVXBPWjNham51UTk0UFI3YU16SDMzblVTQnI
4ZkhZRHJxT0JiNThweEdxSEpSeVgvNk5RTUU0d0hRWURWUjBPQkJZRUZQb0hBM0NMaHhGYkMwSXQ3ekU
0dzhoazVFSi9NQjhHQTFVZEl3UVlNQmFBRlBvSEEzQ0xoeEZiQzBJdDd6RTR3OGhrNUVKL01Bd0dBMVV
kRXdRRk1BTUJBZjh3Q2dZSUtvWkl6ajBFQXdJRFNBQXdSUUloQUowNlFTWHQ5aWhJYkVLWUtJanNQa3J
pVmRMSWd0ZnNiRFN1N0VySmZ6cjRBaUJxb1lDWmYwK3pJNTVhUWVBSGpJekE5WG02M3JydUF4Qlo5cHM
5ejJYTmxRPT0iXSwiaWNvbiI6ImRhdGE6aW1hZ2UvcG5nO2Jhc2U2NCxpVkJPUncwS0dnb0FBQUFOU1V
oRVVnQUFBRThBQUFBdkNBWUFBQUNpd0pmY0FBQUFBWE5TUjBJQXJzNGM2UUFBQUFSblFVMUJBQUN4and
2OFlRVUFBQUFKY0VoWmN3QUFEc01BQUE3REFjZHZxR1FBQUFhaFNVUkJWR2hEN1pyNWJ4UmxHTWY5S3p
UQjhBTS9ZRWhFMlc3cFFaY1dLS0JjbFNwSEFUbEVMQVJFN2tORUNDQTNGa1dLMENLS1NDRklzS0JjZ1Z
DRFdHTkVTZEFZaWR3Z2dnSkJpUmlNaEZjLzR3eTg4ODR6dTlOZGxuR1RmWkpQMm4zbk8rKzg4OTMzZnZ
lQkJ4K1BxQ3pKa1RVdkJiTG1wVURXdkJUSW1wY0NTWnZYTENkWDlSMDVTazE5YmI1YXRmNTk5ZkcrL2V
yQTU0MXE0N2FQMUxMVmE5U0l5Vk5VaThJaThkNWtHVHNpMzBORnY3YWk5bjdRWlBNd2JkeXMyZXJVMlh
NcVVkeTgrWmNhTm1HaW1FOHlYTjNSVWQzYTE4bkYwZlVsb3ZaKzBDVHpXcGQyVmorZU9tMWJFeXk2RHg
0aTVwVU1HV3ZlbzUwNnEyMjdkdHVXQkl1ZmZyNm9XcFYwRlBOTGhvdzE3NTFObTIxTHZQSDNyVnRXamZ
6NjZMZnFsOHRYN0ZSbDlZRlNYc21Tc2ViOWNlT0diWWs3TU5VY0dQZzhac2JNZTlyZlFVYWFWL0pNWDl
zcWR6RENTdnAwa1pIbVRaZzl4N2JMSGNNblRoYjE2ZUorbVZmUXE4eWFVWlFORzY0aVhaKzAva3E2dU9
aRk8wUXRhdGRXS2ZYblJROTlCajkxUjVPSUZuazU0ak4wbWtVaXFsTzNYRFcrTWwrOThtS0I2dFc3cld
wWmNQYyswemc0dExyWWxVYzg2RTZlR0RqSU11YlZwY3VzZWFyZmdJWUdSazZicmhaVnIvSmNIem9vTDc
1NTBqZWRMRXhvcFdjQXBpMlpVcWh1N0pMdnJWc1FVODF6a3pPUGVlbU1SWXZWdVFzWDdQYmlEUVk1SnZ
ab25mdEsrMVZZOEg5dXR4NTMwaDBvYitqbVJZcWo2b3VhWXZFZW5XL1dsWWpwOGN3Yk1tNjgydFB3cVc
xUjR0ai8yU0gxM0lSSllsNG1vWnZYcGlTcURyN2RYdFFIeGEvUEszLytCV3NLMWRUZ0h1NlY4dFFKM2J
3Rmt3cEZyVU9RNTBzMXIzbGV2bTh6WmNxMTcrQkJhdzdLOGxFSzVxemtZZWFyazlBOHA3UDNHekRLK25
kM0RRb3crNlVDOFNWTjgyaXV2MzhpbTdOdGFYdFYxQ1ZxNlJndzRwa3NtYmRpM2J1MkRlN1lmYUJCeGN
xZnZxUHJVakZRTlRRMjJsZmRVVlZUNjhyVEpLRjVEblNtVWpnZHFnNG1TUzlwbXNmREpSM0c2VG9IMGl
XOWFWN0xXTEhZWEtsbFREdDBMVEF0a1lJYWFtcDFRalZ2Kyt1eUdVeFZkSjBETlZYU20rYjFxUnhwbDg
0ZGRmWDFMcDFPL2Q2OXRzb2QwdnM1aEdyZTl4dThvK2ZwTFIxY0doTlRENlo1N0M5S01XWGVmSmRPWjk
0YmI5b3FkMVJPblM3cUlUVHpIaW1NcWl2Yk8zZzBEZFZ5azNXUUJoQnp0SzM1WUtOZE9uYzhPM2FjUzZ
mRFpGZ0thWExzRUpwNXJkcmxpQnFwODljSmNzL203VHZzMHJrakdmTjRiMGtQb1puM1VKdUlPcm5aMjJ
5UDFmbXZVeCtPNWdTcWViVjFtK3pTdVlOVmhxN1RXYkRpTFZ2bGpwbExsb3A2Q0xYUCsycXR2R0xJTC8
xdmltSVNkTUJnelNvRlp5dTZUcWQranp4Z3NQYVY5QkNxZWUvTmpZazZ2NmxLOWN3aVVjL1NUdGYxSER
wTTNiNTkyeTdoM1RoeDVveks2OUhMcFlXdUF3YXFTNWN2MjZxN2NlYjhlZlZZYVJlUDNpRlU4emoxa25
Td1pYSE1tbkNqWTBPZ2FsbzdVUWZTQ00zcVFRcjJIL1hGUDdzc1h4NDVZbDkxQnllQ2VwNG1vWm9IKzF
mRzN4RDR0VDd4OGt3eWo4bndiOWV2MjZWMEI2ZCs3SDR6S3Z1ZEFINTM3RmpxeXpPSGRKbkhFdXptWHE
vV2p4T2J2Tk1idjduaHl3c1gyYVZzV3RDOCs0OGFMZWFwRTdwNXdLWmkwQTJBUVJWNW52UjRFK3VKYyt
iNjFrQXBxSW54QmdtZC80VjVRUC9tdDE4SERDN3NSSGZ0bWV1NWxtaFYwcm4vQUxYMjMyYnFkNEJGbkR
4N1ZpMWNXUzJ1ZmYwSWJCNDdxZXh4bVVqOVF1dFlqdXBkM3RZRDZhYldCQk1yaCthcE5iT0tyTkYxK3V
nQ2E0cmlYR2Z3TVBQdFZpYXZoVTNZTU9BQW51VWIvUjA3TDB5T1NlT2FkRTg4QXBzWEZHZmYzMHluaGx
KZ001MUNVNnZOOUV6Z25wdkhCRlV5aVZyYWVQaXdKNTNERjVaVFpub21FTmc4NWtOVWQyb0ppMldwcjR
PbW1rZk40eDR6SGZpVkZjOER2OE56dWhOcU9pZGlsR3ZBNkRHdWVad083OEFBUW42Y2lFazYrcnc1VmN
2anZxTkRZUE9vSVV3YUtTaHJ4QXVYTGxrSDRhWXVHZk1ZRGMxMFdGNVRhMzFoUEpPZmNVaHJVL0psSU5
pNmM2ZWxSWWRCcG82KytZZmp4NjFsR05mUm00TUQ1ckoxajNGb0dIbmpEU0JOYXJZVWdNTHlNc3pLcGI
3dFhwb0hmUHM4aDNXcDFMek5mTms1NFh4QzF3REdVbVl6WFllZmg2ei9jS3RWbTRFQnhhOVZRR0R6WXI
zTHJVTVJqSEVLa2s3emFGS1lRQTJoR1FVMXorODVORldwWERya3ozdngxMEdxeFE2QnplTmJvQms1bjh
rNG5lYlJoK2sxaFdmeFRGMEQxRXlXVXM1bnYrZGdRcUtheHp1Q2RFMGlzSGwwMk5ROGFoMG1YcjEyTGE
zbTBmOXdpazkrd0xOVE1ZLzg2TVBvOHlpMzFPZnhtVDZQV29xRzkrRFp1a1luYTU2bVNadDVXV1N5NXF
WQTFyd1V5SnFYQWxuemtpYWkvZ0hTRDdSa1R5aWhvZ0FBQUFCSlJVNUVya0pnZ2c9PSJ9LCJzdGF0dXN
SZXBvcnRzIjpbeyJzdGF0dXMiOiJGSURPX0NFUlRJRklFRCIsImVmZmVjdGl2ZURhdGUiOiIyMDE0LTA
xLTA0In1dLCJ0aWJlT2ZMYXN0U3RhdHVzO2hhbmdlIjoiMiAxNC0wMS0wNCJ9LHsiYWEndWlkIjoiMDE
```

```
zMmQxMTAtYmY0ZS00MjA4LWE0MDMtYWI0ZjVmMTJlZmU1IiwibWV0YWRhdGFTdGF0ZW1lbnQiOnsibGV
nYWxIZWFkZXIiOiJodHRwczovL2ZpZG9hbGxpYW5jZS5vcmcvbWV0YWRhdGEvbWV0YWRhdGEtc3RhdGV
tZW50LWxlZ2FsLWhlYWRlci8iLCJkZXNjcmlwdGlvbiI6IkZJRE8gQWxsaWFuY2UgU2FtcGxlIEZJRE8
yIEF1dGhlbnRpY2F0b3IiLCJhYWd1aWQiOiIwMTMyZDExMC1iZjRlLTQyMDgtYTQwMy1hYjRmNWYxMmV
mZTUiLCJhbHRlcm5hdGl2ZURlc2NyaXB0aW9ucyI6eyJydS1SVSI6ItCf0YDQuNC80LXRgCBGSURPMiD
QsNGD0YLQtdC90YLQuNGE0LjQutCw0YLQvtGA0LAg0L7RgiBGSURPIEFsbGlhbmNlIiwiZnItRlIiOiJ
FeGVtcGxlIEZJRE8yIGF1dGhlbnRpY2F0b3IgZGUgRklETyBBbGxpYW5jZSIsInpoLUNOIjoi5L6G6Ie
qRklETyBBbGxpYW5jZeeahOekuuS-i0ZJRE8y6Lqr5Lu96amX6K2J5ZmoIn0sInByb3RvY29sRmFtaWx
5IjoiZmlkbzIiLCJzY2hlbWEiOjMsImF1dGhlbnRpY2F0b3JWZXJzaW9uIjo1LCJ1cHYiOlt7Im1ham9
yIjoxLCJtaW5vciI6MH1dLCJhdXRoZW50aWNhdGlvbkFsZ29yaXRobXMiOlsic2VjcDI1NnIxX2VjZHN
hX3NoYTI1Nl9yYXciLCJyc2Fzc2FfcGtjc3YxNV9zaGEyNTZfcmF3Il0sInB1YmxpY0tleUFsZ0FuZEV
uY29kaW5ncyI6WyJjb3NlIl0sImF0dGVzdGF0aW9uVHlwZXMiOlsiYmFzaWNfZnVsbCJdLCJ1c2VyVmV
yaWZpY2F0aW9uRGV0YWlscyI6W1t7InVzZXJWZXJpZmljYXRpb25NZXRob2QiOiJub25lIn1dLFt7InV
zZXJWZXJpZmljYXRpb25NZXRob2QiOiJwcmVzZW5jZV9pbnRlcm5hbCJ9XSxbeyJ1c2VyVmVyaWZpY2F
0aW9uTWV0aG9kIjoicGFzc2NvZGVfZXh0ZXJuYWwiLCJjYURlc2MiOnsiYmFzZSI6MTAsIm1pbkxlbmd
0aCI6NH19XSxbeyJ1c2VyVmVyaWZpY2F0aW9uTWV0aG9kIjoicGFzc2NvZGVfZXh0ZXJuYWwiLCJjYUR
lc2MiOnsiYmFzZSI6MTAsIm1pbkxlbmd0aCI6NH19LHsidXNlclZlcmlmaWNhdGlvbk1ldGhvZCI6InB
yZXNlbmNlX2ludGVybmFsIn1dXSwia2V5UHJvdGVjdGlvbiI6WyJoYXJkd2FyZSIsInNlY3VyZV9lbGV
tZW50Il0sIm1hdGNoZXJQcm90ZWN0aW9uIjpbIm9uX2NoaXAiXSwiY3J5cHRvU3RyZW5ndGgiOjEyOCw
iYXR0YWNobWVudEhpbnQiOlsiZXh0ZXJuYWwiLCJ3aXJlZCIsIndpcmVsZXNzIiwibmZjIl0sInRjRGl
zcGxheSI6W10sImF0dGVzdGF0aW9uUm9vdENlcnRpZmljYXRlcyI6WyJNSUlDUFRDQ0FlT2dBd0lCQWd
JSkFPdWV4dlUzT3kyd01Bb0dDQ3FHU000OUJBTUNNSHN4SURBZUJnTlZCQU1NRjFOaGJYQnNaU0JCZEh
SbGMzUmhkR2x2YmlCU2IyOTBNUll3RkFZRFZRUUtEQTFHU1VSUElFRnNiR2xoYm1ObE1SRXdEd1lEVlF
RTERBaFZRVVlnVkZkSExERVNNQkFHQTFVRUJ3d0pVR0ZzYnlCQmJIUnZNUXN3Q1FZRFZRUUlEQUpEUVR
FTE1Ba0dBMVVFQmhNQ1ZWTXdIaGNOTVRRd05qRTRNVE16TXpNeVdoY05OREV4TVRBek1UTXpNek15V2p
CN01TQXdIZ1lEVlFRRERCZFRZVzF3YkdVZ1FYUjBaWE4wWVhScGIyNGdVbTl2ZERFV01CUUdBMVVFQ2d
3TlJrbEVUeUJCYkd4cFlXNWpaVEVSTUE4R0ExVUVDd3dJVlVGR0lGUlhSeXd4RWpBUUJnTlZCQWNNQ1Z
CaGJHOGdRV3gwYnpFTE1Ba0dBMVVFQ0F3Q1EwRXhDekFKQmdOVkJBWVRBbFZUTUZrd0V3WUhLb1pJemo
wQ0FRWUlLb1pJemowREFRY0RRZ0FFSDhodjJEMEhYYTU5L0JtcFE3UlplaEwvRk1HekZkMVFCZzl2QVV
wT1ozYWpudVE5NFBSN2FNekgzM25VU0JyOGZIWURycU9CYjU4cHhHcUhKUnlYLzZOUU1FNHdIUVlEVlI
wT0JCWUVGUG9IQTNDTGh4RmJDMEl0N3pFNHc4aGs1RUovTUI4R0ExVWRJd1FZTUJhQUZQb0hBM0NMaHh
GYkMwSXQ3ekU0dzhoazVFSi9NQXdHQTFVZEV3UUZNQU1CQWY4d0NnWUlLb1pJemowRUF3SURTQUF3UlF
JaEFKMDZRU1h0OWloSWJFS1lLSWpzUGtyaVZkTElndGZzYkRTdTdFckpmenI0QWlCcW9ZQ1pmMCt6STU
1YVFlQUhqSXpBOVhtNjNycnVBeEJaOXBzOXoyWE5sUT09Il0sImljb24iOiJkYXRhOmltYWdlL3BuZzt
iYXNlNjQsaVZCT1J3MEtHZ29BQUFBTlNVaEVVZ0FBQUU4QUFBQXZDQVlBQUFDaXdKZmNBQUFBQVhOU1I
wSUFyczRjNlFBQUFBUm5RVTFCQUFDeGp3djhZUVVBQUFBSmNFaFpjd0FBRHNNQUFBN0RBY2R2cUdRQUF
BYWhTVVJCVkdoRDdacjVieFJsR01mOUt6VEI4QU0vWUVoRTJXN3BRWmNXS0tCY2xTcEhBVGxFTEFSRTd
rTkVDQ0EzRmtXSzBDS0tTQ0ZJc0tCY2dWQ0RXR05FU2RBWWlkd2dnZ0pCaVJpTWhGYy80d3k4ODg0enU
5TmRsbkdUZlpKUDJuM25PKys4ODkzM2Z2ZUJCeCtQcUN6SmtUVXZCYkxtcFVEV3ZCVEltcGNDU1p2WEx
DZFg5UjA1U2sxOWJiNWF0ZjU5OWZHKy9lckE1NDFxNDdhUDFMTFZhOVNJeVZOVWk4SWk4ZDVrR1RzaTM
wTkZ2N2FpOW43UVpQTXdiZHlzMmVyVTJYTXFVZHk4K1pjYU5tR2ltRTh5WE4zUlVkM2ExOG5GMGZVbG9
2WiswQ1R6V3BkMlZqK2VPbTFiRXl5NkR4NGk1cFVNR1d2ZW81MDZxMjI3ZHR1V0JJdWZmcjZvV3BWMEZ
QTkxob3cxNzUxTm0yMUx2UEgzclZ0V2pmejY2TGZxbDh0WDdGUmw5WUZTWHNtU3NlYjljZU9HYllrN01
OVWNHUGc4WnNiTWU5cmZRVWFhVi9KTVg5c3FkekRDU3ZwMGtaSG1UWmc5eDdiTEhjTW5UaGIxNmVKK21
WZlFxOHlhVVpRTkc2NGlYWiswL2txNnVPWkZPMFF0YXRkV0tmWG5SUTk5Qmo5MVI1T0lGbms1NGpOMG1
rVWlxbE8zWERXK01sKzk4bUtCNnRXN3JXcFpjUGMrMHpnNHRMcllsVWM4NkU2ZUdEaklNdWJWcGN1c2V
hcmZnSVlHUms2YnJoWlZyL0pjSHpvb0w3NTUwamVkTEV4b3BXY0FwaTJaVXFodTdKTHZyVnNRVTgxemt
6T1BlZW1NUll2VnVRc1g3UGJpRFFZNUp2Wm9uZnRLKzFWWThIOXV0eDUzMGgwb2Iram1SWXFqNm91YVl
2RWVuVy9XbFlqcDhjd2JNbTY4MnRQd3FXMVI0dGovMlNIMTNJUkpZbDRtb1p2WHBpU3FEcjdkWHRRSHh
hL1BLMy8rQldzSzFkVGdIdTZWOHRRSjNid0Zrd3BGclVPUTUwczFyM2xldm04elpjcTE3K0JCYXc3Szh
sRUs1cXprWWVhcms5QThwN1AzR3pESytuZDNEUW93KzZVQzhTVk44Mml1djM4aW03TnRhWHRWMUNWcTZ
SZ3c0cGtzbWJkaTNidTJEZTdZZmFCQnhjcWZ2cVByVWpGUU5UUTIybGZkVVZWVDY4clRKS0Y1RG5TbVV
qZ2RxZzRtU1M5cG1zZkRKUjNHNlRvSDBpVzlhVjdMV0xIWVhLbGxURHQwTFRBdGtZSWFhbXAxUWpWdis
rdXlHVXhWZEowRE5WWFNtK2IxcVJ4cGw4NGRkZlgxTHAxTy9kNjl0c29kMHZzNWhHcmU5eHU4bytmcEx
SMWNHaE5URDZaNTdDOUtNV1hlZkpkT1o5NGJiOW9xZDFST25TN3FJVFR6SGltTXFpdmJPM2cwRGRWeWs
zV1FCaEJ6dEszNVlLTmRPbmM4TzNhY1M2ZkRaRmdLYVhMc0VKcDVyZHJsaUJxcDg5Y0pjcy9tN1R2czB
ya2pHZk40YjBrUG9abjNVSnVJT3JuWjIyeVAxZm12VXgrTzVnU3FlYlYxbSt6U3VZTlZocTdUV2JEaUx
WdmxqcGxMbG9wNkNMWFArMnF0dkdMSUwvMXZpbUlTZE1CZ3pTb0ZaeXU2VHFkK2p6eGdzUGFWOUJDcWV
lL05qWWs2djZsSzljd2lVYy9TVHRmMUhEcE0zYjU5Mnk3aDNUaHg1b3pLNjlITHBZV3VBd2FxUzVjdjI
2cTdjZWI4ZWZWWWFSZVAzaUZVOHpqMWtuU3daWEhNbW5DalkwT2dhbG83VVFmU0NNM3FRUXIySC9YRlA
3c3NYeDQ1WWw5MUJ5ZUNlcDRtb1pvSCsxZkczeEQ0dFQ3eDhrd3lqOG53YjlldjI2VjBCNmQrN0g0ekt
2dWRBSDUzN0ZqcXl6T0hkSm5IRXV6bVhxL1dqeE9idk5NYnY3bmh5d3NYMmFWc1d0QzgrNDhhTGVhcEU
```

```
3cDV3S1ppMEEyQVFSVjVudlI0RSt1SmMrYjYxa0FwcUlueEJnbWQvNFY1UVAvbXQxOEhEQzdzUkhmdG1
ldTVsbWhWMHJuL0FMWDIzMmJxZDRCRm5EeDdWaTFjV1MydWZmMEliQjQ3cWV4eG1VajlRdXRZanVwZDN
0WUQ2YWJXQkJNcmgrYXBOYk9Lck5GMSt1Z0NhNHJpWEdmd01QUHRWaWF2aFUzWU1PQUFudVViL1IwN0w
weU9TZU9hZEU4OEFwc1hGR2ZmMzB5bmhsSmdNNTFDVTZ2TjlFemducHZIQkZVeWlWcmFlUGl3SjUzREY
1WlRabm9tRU5nODVrTlVkMm9KaTJXcHI0T21ta2ZONHg0ekhmaVZGYzhEdjhOenVoTnFPaWRpbEd2QTZ
ER3VlWndPNzhBQVFuNmNpRWs2K3J3NVZjdmp2cU5EWVBPb0lVd2FLU2hyeEF1WExsa0g0YVl1R2ZNWUR
jMTBXRjVUYTMxaFBKT2ZjVWhyVS9KbElOaTZjNmVsUllkQnBvNisrWWZqeDYxbEdOZlJtNE1ENXJKMWo
zRm9HSG5qRFNCTmFyWVVnTUx5TXN6S3BiN3RYcG9IZlBzOGgzV3AxTHpOZk5rNTRYeEMxd0RHVW1Zelh
ZZWZoNnovY0t0Vm00RUJ4YTlWUUdEellyM0xyVU1SakhFS2trN3phRktZUUEyaEdRVTF6Kzg1TkZXcFh
Ecmt6M3Z4MTBHcXhRNkJ6ZU5ib0JrNW44azRuZWJSaCtrMWhXZnhURjBEMUV5V1VzNW52K2RnUXFLYXh
6dUNkRTBpc0hsMDJOUThhaDBtWHIxMkxhM20wZjl3aWs5K3dMTlRNWS84Nk1Qbzh5aTMxT2Z4bVQ2UFd
vcUc5K0RadWtZbmE1Nm1TWnQ1V1dTeTVxVkExcndVeUpxWEFsbnpraWFpL2dIU0Q3UmtUeWlob2dBQUF
BQkpSVTVFcmtKZ2dnPT0iLCJzdXBwb3J0ZWRFeHRlbnNpb25zIjpbeyJpZCI6ImhtYWMtc2VjcmV0Iiw
iZmFpbF9pZl91bmtub3duIjpmYWxzZX0seyJpZCI6ImNyZWRQcm90ZWN0IiwiZmFpbF9pZl91bmtub3d
uIjpmYWxzZX1dLCJhdXRoZW50aWNhdG9yR2V0SW5mbyI6eyJ2ZXJzaW9ucyI6WyJVMkZfVjIiLCJGSUR
PXzJfMCJdLCJleHRlbnNpb25zIjpbImNyZWRQcm90ZWN0IiwiaG1hYy1zZWNyZXQiXSwiYWFndWlkIjo
iMDEzMmQxMTBiZjRlNDIwOGE0MDNhYjRmNWYxMmVmZTUiLCJvcHRpb25zIjp7InBsYXQiOiJmYWxzZSI
sInJrIjoidHJ1ZSIsImNsaWVudFBpbiI6InRydWUiLCJ1cCI6InRydWUiLCJ1diI6InRydWUiLCJ1dlR
va2VuIjoiZmFsc2UiLCJjb25maWciOiJmYWxzZSJ9LCJtYXhNc2dTaXplIjoxMjAwLCJwaW5VdkF1dGh
Qcm90b2NvbHMiOlsxXSwibWF4Q3JlZGVudGlhbENvdW50SW5MaXN0IjoxNiwibWF4Q3JlZGVudGlhbEl
kTGVuZ3RoIjoxMjgsInRyYW5zcG9ydHMiOlsidXNiIiwibmZjIl0sImFsZ29yaXRobXMiOlt7InR5cGU
iOiJwdWJsaWMta2V5IiwiYWxnIjotN30seyJ0eXBlIjoicHVibGljLWtleSIsImFsZyI6LTI1N31dLCJ
tYXhBdXRoZW50aWNhdG9yQ29uZmlnTGVuZ3RoIjoxMDI0LCJkZWZhdWx0Q3JlZFByb3RlY3QiOjIsImZ
pcm13YXJlVmVyc2lvbiI6NX19LCJzdGF0dXNSZXBvcnRzIjpbeyJzdGF0dXMiOiJGSURPX0NFUlRJRkl
FRCIsImVmZmVjdGl2ZURhdGUiOiIyMDE5LTAxLTA0In0seyJzdGF0dXMiOiJGSURPX0NFUlRJRklFRF9
MMSIsImVmZmVjdGl2ZURhdGUiOiIyMDIwLTExLTE5IiwiY2VydGlmaWNhdGlvbkRlc2NyaXB0b3IiOiJ
GSURPIEFsbGlhbmNlIFNhbXBsZSBGSURPMiBBdXRoZW50aWNhdG9yIiwiY2VydGlmaWNhdGVOdW1iZXI
iOiJGSURPMjEwMDAyMDE1MTIyMTAwMSIsImNlcnRpZmljYXRpb25Qb2xpY3lWZXJzaW9uIjoiMS4wLjE
iLCJjZXJ0aWZpY2F0aW9uUmVxdWlyZW1lbnRzVmVyc2lvbiI6IjEuMC4xIn1dLCJ0aW1lT2ZMYXN0U3R
hdHVzQ2hhbmdlIjoiMjAxOS0wMS0wNCJ9XX0.-kc1wrorJA16bxLXXzeDkFEOCsbKAy2WDEzoCY-Aej_
N0bWIOAmhpHGxSa3CXgmwFwgAuy230Eq_BHTO_RshsA
```

The line breaks are for display purposes only.

The signature in the example above was computed with the following ECDSA key

```
EXAMPLE: ECDSA KEY USED FOR SIGNATURE COMPUTATION
-----BEGIN CERTIFICATE----
MIICZTCCAgugAwIBAgIBATAKBggqhkjOPQQDAjCBozEnMCUGA1UEAwweRVhBTVBM
RSBNRFMzIFRFU1QgSU5URVJNRURJQVRFMSIwIAYJKoZIhvcNAQkBFhNleGFtcGxl
QGV4YW1wbGUuY29tMRQwEgYDVQQKDAtFeGFtcGxlIE9SRzEQMA4GA1UECwwHRXhh
bXBsZTELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk1ZMRIwEAYDVQQHDAlXYWtlZmll
bGQwHhcNMjEwNDE5MTEzNTA3WhcNMzEwNDE3MTEzNTA3WjCBpTEpMCcGA1UEAwwg
RVhBTVBMRSBNRFMzIFNJR05JTkcgQ0VSVElGSUNBVEUxIjAgBgkqhkiG9w0BCQEW
E2V4YW1wbGVAZXhhbXBsZS5jb20xFDASBgNVBAoMC0V4YW1wbGUgT1JHMRAwDgYD
VQQLDAdFeGFtcGxlMQswCQYDVQQGEwJVUzELMAkGA1UECAwCTVkxEjAQBgNVBAcM
CVdha2VmaWVsZDBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IABNQJs6wTqixc+S+V
DAajFlPNat10KEWJE5jcWOvm6qpO9SDAAMZvb4HHrvs+P5YRpHrSlUPdvK+uEQbd
Wg31P9ujLDAqMAkGA1UdEwQCMAAwHQYDVR0OBBYEFLqsapcXV4ZoVHAnRpPZwQe7
Yy20MAoGCCqGSM49BAMCA0gAMEUCIQC67za8EIuyRiKgNDXIP1s1aLr3jzH9WVXf
Hx4bJ+zCsgIgG/tVButOJUU+vvoHIo/otAUAcH5bNHP3uIziDS+PTUc=
-----END CERTIFICATE----   -----BEGIN EC PRIVATE KEY----
MHcCAQEEIFNpFhJvod3jKvbrLLzKTWKFxzaZ4l7kMchx3NyytQYUoAoGCCqGSM49
AwEHoUQDQgAE1AmzrBOqLFz5L5UMBqMWU81q3XQoRYkTmNxY6+bqqk71IMAAxm9v
gceu+z4/lhGketKVQ928r64RBt1aDfU/2w==
-----END EC PRIVATE KEY----
```

The root certificate to validate certificate path in the X5C is:

```
EXAMPLE: CERTIFICATE PATH ROOT CERTIFICATE
-----BEGIN CERTIFICATE-----
MIIGGTCCBAGgAwIBAgIUdT9qLX0sVMRe8l0sLmHd3mZovQ0wDQYJKoZIhvcNAQEL
BQAwgZsxHzAdBgNVBAMMFkVYQU1QTEUgTURTMyBURVNUIFJPT1QxIjAgBgkqhkiG
9w0BCQEWE2V4YW1wbGVAZXhhbXBsZS5jb20xFDASBgNVBAoMC0V4YW1wbGUgT1JH
MRAwDgYDVQQLDAdFeGFtcGxlMQswCQYDVQQGEwJVUzELMAkGA1UECAwCTVkxEjAQ
BgNVBAcMCVdha2VmaWVsZDAeFw0yMTA0MTkxMTM1MDdaFw00ODA5MDQxMTM1MDda
MIGbMR8wHQYDVQQDDBZFWEFNUExFIE1EUzMgVEVTVCBST09UMSIwIAYJKoZIhvcN
AQkBFhNleGFtcGxlQGV4YW1wbGUuY29tMRQwEgYDVQQKDAtFeGFtcGxlIE9SRzEQ
MA4GA1UECwwHRXhhbXBsZTELMAkGA1UEBhMCVVMxCzAJBgNVBAgMAk1ZMRIwEAYD
VQQHDAlXYWtlZmllbGQwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoICAQDD
jF5wyEWuhwDHsZosGdGFTCcI677rW881vV+UfW38J+K2ioFFNeGVsxbcebK6AVOi
CDPFj0974IpeD9SFOhwAHoDu/LCfXdQWp8ZgQ91ULYWoW8o7NNSp01nbN9zmaO6/
xKNCa0bzjmXoGqglqnP1AtRcWYvXOSKZy1rcPeDv4Dhcpdp6W72fBw0eWIqOhsrI
tuY2/N8ItBPiG03EX72nACq4nZJ/nAIcUbER8STSFPPzvE97TvShsi1FD8aO6l1W
kR/QkreAGjMI++GbB2Qc1nN9Y/VEDbMDhQtxXQRdpFwubTjejkN9hKOtF3B71Yrw
Irng3V9RoPMFdapWMzSlI+WWHog0oTj1PqwJDDg7+z1I6vSDeVWAMKr9mq1w1OGN
zgBopIjd9lRWkRtt2kQSPX9XxqS4E1gDDr8MKbpM3JuubQtNCg9D7Ljvbz6vwvUr
bPHH+oREvucsp0PZ5PpizloepGIcLFxDQqCulGY2n7Ahl0JOFXJqOFCaK3TWHwBv
ZsaY5DgBuUvdUrwtgZNg2eg2omWXEepiVFQn3Fvj43Wh2npPMgIe5P0rwncXvROx
aczd4rtajKS1ucoB9b9iKqM2+M1y/FDIgVf1fWEHwK7YdzxMlgOeLdeV/kqRU5PE
UlLU9a2EwdOErrPbPKZmIfbs/L4B3k4zejMDH3Y+ZwIDAQABo1MwUTAdBgNVHQ4E
FgQU8sWwq1TrurK7xMTwO1dKfeJBbCMwHwYDVR0jBBgwFoAU8sWwq1TrurK7xMTw
O1dKfeJBbCMwDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAgEAFw6M
1PiIfCPIBQ5EBUPNmRvRFuDpolOmDofnf/+mv63LqwQZAdo/W8tzZ9kOFhq24SiL
w0H7fsdG/jeREXiIZMNoW/rA6Uac8sU+FYF7Q+qp6CQLlSQbDcpVMifTQjcBk2xh
+aLK9SrrXBqnTAhwS+offGtAW8DpoLuH4tAcQmIjlgMlN65jnELCuqNR/wpA+zch
8LZW8saQ2cwRCwdr8mAzZoLbsDSVCHxQF3/kQjPT7Nao1q2iWcY3OYcRmKrieHDP
67yeLUbVmetfZis2d6ZlkqHLB4ZW1xX4otsEFkuTJA3HWDRsNyhTwx1YoCLsYut5
Zp0myqPNBq28w6qGMyyoJN0Z4RzMEO3R6i/MQNfhK55/8O2HciM6xb5t/aBSuHPK
lBDrFWhpRnKYkaNtlUo35qV5IbKGKau3SdZdSRciaXUd/p81YmoF01UlhhMz/Rqr
1k2gyA0a9tF8+awCeanYt5izl8YO0FlrOU1SQ5UQw4szqqZqbrf4e8fRuU2TXNx4
zk+ImE7WRB44f6mSD746ZCBRogZ/SA5jUBu+OPe4/sEtERWRcQD+fXgce9ZEN0+p
eyJIKAsl5Rm2Bmgyg5IoyWwSG5W+WekGyEokpslou2Yc6EjUj5ndZWz5EiHAiQ74
hNfDoCZIxVVLU3Qbp8a0S1bmsoT2JOsspIbtZUg=
-----END CERTIFICATE-----
```

3.2. Metadata BLOB object processing rules§

The FIDO Server MUST follow these processing rules:

1. Download and cache the root signing trust anchor from the respective MDS root location e.g. More information can be found at https://fidoalliance.org/metadata/
2. To validate the digital certificates used in the digital signature, the certificate revocation information MUST be available in the form of CRLs at the respective MDS CRL location e.g. More information can be found at https://fidoalliance.org/metadata/
3. The FIDO Server MUST be able to download the latest metadata BLOB object from the well-known URL when appropriate, e.g. https://mds.fidoalliance.org/. The `nextUpdate` field of the Metadata BLOB specifies a date when the download SHOULD occur at latest.
4. If the `x5u` attribute is present in the JWT Header, then:
    1. The FIDO Server MUST verify that the URL specified by the `x5u` attribute has the same web-origin as the URL used to download the metadata BLOB from. The FIDO Server SHOULD ignore the file if the web-origin differs (in order to prevent loading objects from arbitrary sites).
    2. The FIDO Server MUST download the certificate (chain) from the URL specified by the `x5u` attribute [JWS]. The certificate chain MUST be verified to properly chain to the metadata BLOB signing trust

anchor according to [RFC5280]. All certificates in the chain MUST be checked for revocation according to [RFC5280].

3. The FIDO Server SHOULD ignore the file if the chain cannot be verified or if one of the chain certificates is revoked.

   > The requirements for verifying certificate revocation, are only applicable to the MDS BLOB payload certificates. It is up to the server vendors whether to enforce CRL check for the certificates in the individual metadata statements.

5. If the `x5u` attribute is missing, the chain should be retrieved from the`x5c` attribute. If that attribute is missing as well, Metadata BLOB signing trust anchor is considered the BLOB signing certificate chain.
6. Verify the signature of the Metadata BLOB object using the BLOB signing certificate chain (as determined by the steps above). The FIDO Server SHOULD ignore the file if the signature is invalid. It SHOULD also ignore the file if its number (`no`) is less or equal to the number of the last Metadata BLOB object cached locally.
7. Write the verified object to a local cache as required.
8. Iterate through the individual entries (of type `MetadataBLOBPayloadEntry`). For each entry:
   1. Ignore the entry if the AAID, AAGUID or attestationCertificateKeyIdentifiers is not relevant to the relying party (e.g. not acceptable by any policy)
   2. Check whether the status report of the authenticator model has changed compared to the cached entry by looking at the fields `timeOfLastStatusChange` and `statusReport`.

      Update the status of the cached entry. It is up to the relying party to specify behavior for authenticators with status reports that indicate a lack of certification, or known security issues. However, the status `REVOKED` indicates significant security issues related to such authenticators.

      > Authenticators with an unacceptable status should be marked accordingly.This information is required for building registration and authentication policies included in the registration request and the authentication request [UAFProtocol].

   3. Update the cached metadata statement.

4. Considerations§

This section is not normative.

This section describes the key considerations for designing this metadata service.

Need for Authenticator Metadata

When defining policies for acceptable authenticators, it is often better to describe the required authenticator characteristics in a generic way than to list individual authenticator AAIDs. The metadata statements provide such information. Authenticator metadata also provides the trust anchor required to verify attestation objects.

The metadata service provides a standardized method to access such metadata statements.

Integrity and Authenticity

Metadata statements include information relevant for the security. Some business verticals might even have the need to document authenticator policies and trust anchors used for verifying attestation objects for auditing purposes.

It is important to have a strong method to verify and proof integrity and authenticity and the freshness of metadata statements. We are using a single digital signature to protect the integrity and authenticity of the Metadata BLOB object and all metadata statements.

Organizational Impact

The FIDO Alliance has control over the FIDO certification process and authentication vendors provide the metadata as part of that process. With this metadata service, the list of known authenticators and their metadata statements need to be updated, signed and published regularly. A single signature needs to be generated in order to protect the integrity and authenticity of the metadata BLOB object and all embedded metadata statements.

Performance Impact

Metadata BLOB objects and metadata statements can be cached by the FIDO Server.

The update policy can be specified by the relying party.

The metadata BLOB object includes a date for the next scheduled update. As a result there is*no additional impact* to the FIDO Server during FIDO Authentication or FIDO Registration operations.

High Security Environments

Some high security environments might only trust internal policy authorities. FIDO Servers in such environments could be restricted to use metadata BLOB objects from a proprietary trusted source only. The metadata service is the baseline for most relying parties.

Extended Authenticator Information

Some relying parties might want additional information about authenticators before accepting them. The policy configuration is under control of the relying party, so it is possible to only accept authenticators for which additional data is available and meets the requirements.

Index§

Terms defined by this specification§

dict-member for StatusReport

certLevel

date

effectiveDate

dict-member for BiometricStatusReport

dict-member for StatusReport

entries

"FIDO_CERTIFIED"

"FIDO_CERTIFIED_L1"

"FIDO_CERTIFIED_L1plus"

"FIDO_CERTIFIED_L2"

"FIDO_CERTIFIED_L2plus"

"FIDO_CERTIFIED_L3"

"FIDO_CERTIFIED_L3plus"

legalHeader

MetadataBLOBPayload

MetadataBLOBPayloadEntry

metadataStatement

modality

nextUpdate

no

"NOT_FIDO_CERTIFIED"

"REVOKED"

RogueListEntry

rogueListHash

rogueListURL

"SELF_ASSERTION_SUBMITTED"

sk

status

StatusReport

statusReports

timeOfLastStatusChange

"UPDATE_AVAILABLE"

url

"USER_KEY_PHYSICAL_COMPROMISE"

"USER_KEY_REMOTE_COMPROMISE"

"USER_VERIFICATION_BYPASS"

Terms defined by reference§

[webauthn-1] defines the following terms:

AAGUID

[WebIDL] defines the following terms:

DOMString

unsigned long

unsigned short

References§

Normative References§

[FIDOAuthenticatorSecurityRequirements]
Rolf Lindemann; Dr. Joshua E. Hill; Douglas Biggs. FIDO Authenticator Security Requirements. November 2020. Final Draft. URL: https://fidoalliance.org/specs/fido-security-requirements/fido-authenticator-security-requirements-v1.4-fd-20201102.html

[FIDOBiometricsRequirements]
Stephanie Schuckers; et al. FIDO Biometrics Requirements. October 2020. URL: https://fidoalliance.org/specs/biometric/requirements/Biometrics-Requirements-v2.0-fd-20201006.html

[FIDOMetadataStatement]
B. Hill; D. Baghdasaryan; J. Kemp. FIDO Metadata Statements. Implementation Draft. URL: https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-metadata-statement-v2.0-id-20180227.html

[JWS]
M. Jones; J. Bradley; N. Sakimura. JSON Web Signature (JWS). May 2015. RFC. URL: https://tools.ietf.org/html/rfc7515

[JWT]
M. Jones; J. Bradley; N. Sakimura. JSON Web Token (JWT). May 2015. RFC. URL: https://tools.ietf.org/html/rfc7519

[RFC4648]
S. Josefsson. The Base16, Base32, and Base64 Data Encodings (RFC 4648). October 2006. URL: http://www.ietf.org/rfc/rfc4648.txt

[RFC5280]
D. Cooper; et al. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. May 2008. URL: https://tools.ietf.org/html/rfc5280

[WEBAUTHN-1]
Dirk Balfanz; et al. Web Authentication:An API for accessing Public Key Credentials Level 1. 4 March 2019. REC. URL: https://www.w3.org/TR/webauthn-1/

[WebIDL]
Boris Zbarsky. Web IDL. 15 December 2016. ED. URL: https://heycam.github.io/webidl/

[WebIDL-ED]
Cameron McCormack. Web IDL. 13 November 2014. Editor's Draft. URL: http://heycam.github.io/webidl/

Informative References§

[FIDOEcdaaAlgorithm]
R. Lindemann; et al. FIDO ECDAA Algorithm. Implementation Draft. URL: https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-ecdaa-algorithm-v2.0-id-20180227.html

[FIDOGlossary]
R. Lindemann; et al. FIDO Technical Glossary. Implementation Draft. URL: https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-glossary-v2.0-id-20180227.html

[FIDOKeyAttestation]
FIDO 2.0: Key attestation format. URL: https://fidoalliance.org/specs/fido-v2.0-ps-20150904/fido-key-attestation-v2.0-ps-20150904.html

[ITU-X690-2008]
X.690: Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER),

Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), (T-REC-X.690-200811). November 2008. URL: https://www.itu.int/rec/T-REC-X.690-200811-S

[RFC2119]
S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. March 1997. Best Current Practice. URL: https://tools.ietf.org/html/rfc2119

[UAFProtocol]
R. Lindemann; et al. FIDO UAF Protocol Specification v1.2. Proposed Standard. URL: https://fidoalliance.org/specs/fido-uaf-v1.2-ps-20201020/fido-uaf-protocol-v1.2-ps-20201020.html

IDL Index§

```
dictionary MetadataBLOBPayloadEntry {
    AAID                    aaid;
    AAGUID                  aaguid;
    DOMString[]             attestationCertificateKeyIdentifiers;
    MetadataStatement       metadataStatement;
    BiometricStatusReport[] biometricStatusReports;
    required StatusReport[] statusReports;
    required DOMString      timeOfLastStatusChange;
    DOMString               rogueListURL;
    DOMString               rogueListHash;
};

dictionary BiometricStatusReport {
    required unsigned short certLevel;
    required DOMString      modality;
    DOMString               effectiveDate;
    DOMString               certificationDescriptor;
    DOMString               certificateNumber;
    DOMString               certificationPolicyVersion;
    DOMString               certificationRequirementsVersion;
};

dictionary StatusReport {
    required AuthenticatorStatus status;
    DOMString                    effectiveDate;
    unsigned long                authenticatorVersion;
    DOMString                    certificate;
    DOMString                    url;
    DOMString                    certificationDescriptor;
    DOMString                    certificateNumber;
    DOMString                    certificationPolicyVersion;
    DOMString                    certificationRequirementsVersion;
};

enum AuthenticatorStatus {
    "NOT_FIDO_CERTIFIED",
    "FIDO_CERTIFIED",
    "USER_VERIFICATION_BYPASS",
    "ATTESTATION_KEY_COMPROMISE",
    "USER_KEY_REMOTE_COMPROMISE",
    "USER_KEY_PHYSICAL_COMPROMISE",
    "UPDATE_AVAILABLE",
    "REVOKED",
    "SELF_ASSERTION_SUBMITTED",
    "FIDO_CERTIFIED_L1",
    "FIDO_CERTIFIED_L1plus",
    "FIDO_CERTIFIED_L2",
    "FIDO_CERTIFIED_L2plus",
```

```
    "FIDO_CERTIFIED_L3",
    "FIDO_CERTIFIED_L3plus"
};

dictionary RogueListEntry {
    required DOMString sk;
    required DOMString date;
};

dictionary MetadataBLOBPayload {
    DOMString                           legalHeader;
    required Number                     no;
    required DOMString                  nextUpdate;
    required MetadataBLOBPayloadEntry[] entries;
};
```

↑

→