FIDO Authenticator Security Requirements version DV 1.2.0 - Level 1 (>=L1)

Section	SR No.	U2F/UAF/ FIDO2,L	Description	Calibration	Vendor Instructions	Vendor Response	L1 Test Procedure	Security Secretariat Response and Results
Authenticator Definition and Derived Authenticator Requirements	1.1	UAF + U2F + FIDO2; >=L1	The vendor SHALL document an explicit Authenticator boundary. The Authenticator's boundary SHALL include any hardware that performs or software that implements functionality used to fulfill FIDO Authenticator Security Requirements, or FIDO Relevant user verification, key generation, secure transaction confirmation display, or signature generation. If the Authenticator includes a software component, the boundary SHALL contain the processor that executes this software. If Transaction Confirmation Display is supported and the Metadata Statement related to this Authenticator claims Transaction Confirmation Display support with tcDisplay including the flag TRANSACTION_CONFIRMATION_DISPLAY_PRIVILEGED_SOFTWARE (0x0002), then the Transaction Confirmation Display MAY be implemented outside of an AROE - even when the Authenticator aims for a certification at L2 and higher. However, in such case the vendor SHALL document where and how Transaction Confirmation Display is implemented. The Authenticator boundary as defined by FIDO is comprised of the hardware and software where the Authenticator runs. The Authenticator Application by definition, is always inside the authenticator boundary. The vendor MUST describe the operational environment for the Authenticator Application, including any specific hardware or operating system requirements to completely define this boundary. The Authenticator always comprises hardware and software and the vendor SHALL describe the boundary. An Authenticator typically belongs to one of the 4 categories: 1- Authenticator Application running on some HLOS without an effective	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, the Authenticator vendor SHALL declare and describe to which of the above mentioned categories the Authenticator Application belongs. At L1, the vendor SHALL also describe what portions of functionality the Authenticator uses from any underlying operating environment that belongs to the Authenticator but that is not included in the Authenticator Application.	This Authenticator belongs to Category 3- as #2, but having the Secret Authenticator Security Parameters protected by an AROE. This Authenticator : - Name of the authenticator: VoiceAuth - Hardware Type & Version: Microphone, SecureElement with TRNG, BLE chip with an external button to activate the BLE, USB port for firmware update. - Underlying Software Platform/OS: XYZ firmwire Transaction Display is NOT implemented. Please refer to the logical representation of Authenticator boundary on the Device sheet.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Authenticator Definition and Derived Authenticator Requirements	1.2	UAF + U2F + FIDO2; >=L1	The vendor SHALL document all FIDO Relevant security and cryptographic functions implemented within the Authenticator, both those on the "Allowed Cryptography List" [FIDOAllowedCrypto] and those not on this list. **Note** Some algorithms may only be allowed for certain Security Certification Levels. For example, not all cryptographic algorithms that are acceptable for L1 may be acceptable for L3.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, the vendor SHALL mark the FIDO Relevant security and cryptographic functions implemented in the Authenticator but implemented outside the Authenticator Application (i.e. in the underlying OS or HW).	AES-GCM128 for key protection & authentication, SHA-256 for Hash Algorithm, Hardware TRNG for Random Number (Fips140 2 certified), ECDSA on P-256 for Attestation Key Pair AES128 for symmetric encryption HMAC-SHA256 for data authentication	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Authenticator Definition and Derived Authenticator Requirements	1.3	UAF + U2F + FIDO2; >=L1	The vendor SHALL document where Authenticator User Private Keys (Uauth.priv) are stored, the structure of all KeyIDs/CredentialIDs and Key Handles used by the Authenticator, and explain how these private keys are related to the KeyIDs/CredentialIDs and Key Handles used by the Authenticator.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, the private keys, KeyIDs/CredentialIDs etc. that are generated outside the Authenticator Application SHALL be documented, but their internal structure does not need to be explained in detail.	**User Private keys:** stored in the RawKeyHandle **Private key's relationship with keyhandle**: UVHash = SHA256 of (User Verification Template(UV)) RawkeyHandle = AES128 (SHA256(KeyID) + Uauth.priv) + KeyID Key Handle = AES128-GCM(RawKeyHandle + Counter + UVhash + SHA256(AppID))	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	

Authenticator Definition and Derived Authenticator Requirements	1.4	UAF + FIDO2; >=L1	The vendor SHALL document an Authenticator as a first-factor Authenticator or a second-factor Authenticator. [UAFAuthnrCommands], [Section 6.3.4] and [FIDOGlossary] entries "Authenticator, 1stF / First Factor" and "Authenticator, 2ndF / Second Factor".	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, in addition to the rationale provided by the vendor, this requirement MUST be demonstrated to the Test Proctor during Interoperability Testing. Documentation is not required.	This authenticator is a First-factor Authenticator	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	
Authenticator Definition and Derived Authenticator Requirements	1.5	UAF + FIDO2; >=L1	If the Authenticator is a second-factor Authenticator, then the Authenticator SHALL NOT store user names (UAF) / PublicKeyCredentialUserEntity (FIDO2) inside a Raw Key Handle [UAFAuthnrCommands], [Section 5.1]. A cryptographically wrapped Raw Key Handle is called Key Handle.	No calibration required	Is this requirement applicable to the Authenticator? If No, then describe why. If Yes, Provide the Security Secretariat with a description of how the requirement above is met.	N/A because it is a first-factor authenticator.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Authenticator Definition and Derived Authenticator Requirements	1.6	UAF + FIDO2; >=L1	Supporting Transaction Confirmation is OPTIONAL for Authenticators. If the Authenticator supports Transaction Confirmation Display, then it SHALL hash the Transaction Content using an Allowed Hashing Cryptographic Function ([UAFAuthnrCommands] Section 6.3.4, [WebAuthn] Section 10.2 and 10.3).	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	This device supports Transaction Confirmation Display and the content of every transaction is hashed using SHA256.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Authenticator Definition and Derived Authenticator Requirements	1.7	UAF + FIDO2; >=L1	If the Authenticator uses the KHAccessToken method of binding keys to apps, then when responding to a “Register”, “Sign”, or “Deregister” command which includes the AppID/RP ID, the Authenticator SHALL use an Allowed Hashing or Data Authentication Cryptographic Function to mix the ASM-provided KHAccessToken and AppID/RP ID. If the Authenticator uses an alternative method of binding keys to apps, the vendor SHALL describe why this method provides equivalent security. Equivalent security means, (1) it prevents other apps (not originating from the same RP) from using the key and (2) in the case of bound Authenticators, it prevents other FIDO Clients of triggering the use of that key, and (3) it may rely on the underlying HLOS platform to work as expected.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	This authenticator implementation uses SHA256 to verify the AppID thereby preventing the use of a key that is not linked to the correct Relying party.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Authenticator Definition and Derived Authenticator Requirements	1.9	UAF + FIDO2; >=L1	Supporting Transaction Confirmation is OPTIONAL for Authenticators. If the Authenticator supports Transaction Confirmation Display, then it SHALL display the transaction content supplied in the “Sign” command. [UAFAuthnrCommands], Section 6.3.4, [FIDOGlossary], and [WebAuthn] Sections 10.2 and 10.3. If the Metadata Statement related to this Authenticator claims Transaction Confirmation Display support with tcDisplay including the flag TRANSACTION_CONFIRMATION_DISPLAY_PRIVILEGED_SOFTWARE (0x0002), the Transaction Confirmation Display MAY be implemented outside of an AROE. If tcDisplay includes the flag TRANSACTION_CONFIRMATION_DISPLAY_TEE, or TRANSACTION_CONFIRMATION_DISPLAY_HARDWARE, then the Transaction Confirmation Display SHALL be implemented inside the AROE as part of the Authenticator.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, in addition to the rationale provided by the vendor, this requirement MUST be demonstrated to the Test Proctor during Interoperability Testing. Documentation is not required.	Authenticator supports Transaction Confirmation Display with metadata flag TRANSACTION_CONFIRMATION_DISPLAY_PRIVILEGED_SOFTWARE. The transaction content displayed is the content supplied in the "sign" command.	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	

Authenticator Definition and Derived Authenticator Requirements	1.10	UAF + U2F + FIDO2; >=L1	Authenticators SHALL validate data input to the Authenticator to defend against buffer overruns, stack overflows, integer under/overflow or other such invalid input-based attack vectors. Note At L2, L3 and L3+ the entire AROE is likely to be within the authenticator boundary and thus part of the Authenticator. Examples of inputs directly related to the FIDO authenticator are FIDO protocol messages and FIDO authenticator configuration inputs. Examples of inputs to the AROE that are not directly related to FIDO are calls to configure the AROE itself or get status from the AROE itself. if the AROE can load and run an application like a signed ELF file, that signed ELF file is an input to the authenticator and the code for verifying and loading the ELF file are subject to this requirement. This is because a malicious ELF file could allow an attacker to compromise the AROE kernel and thus compromise FIDO code running on the AROE. At L2, L3 and L3+ the inputs to the Authenticator are primarily inputs that come from the less-secure or non-secure world outside the AROE. These are typically calls that come from the High-Level or Rich OS. Inputs between modules and subsystems within the AROE are not considered inputs for this requirement. Data read by the AROE from unsecured storage is also considered an input to the AROE.	L1: At L1, the Authenticator Application needs to verify only the inputs to the Authenticator Application before they are processed further by the underlying operating environment.	Provide the Security Secretariat with a rationale of how the requirement above is met.	This authenticator implements input validation (eg. Type-length checks, etc), to defend against input based attacks.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.1	UAF + U2F + FIDO2; >=L1	The vendor SHALL document all Authenticator Security Parameters (ASPs). Data parameters used by or stored within the Authenticator which are FIDO Relevant are called Authenticator Security Parameter. These SHALL, at minimum, include all FIDO user verification reference data, FIDO biometric data, Key Handle Access Tokens, User Verification Tokens (see [UAFAuthnrCommands], Section 5.3 and [FIDOGlossary]), signature or registration operation counters, FIDO Relevant cryptographic keys, and FIDO relevant Allowed Random Number Generator state data. Biometric data is defined as raw captures off the sensor, stored templates, candidate match templates, and any intermediate forms of biometric data. Biometric data not used with FIDO is excluded.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	Please refer to "ASP's Table" for the detailed documentation of all ASPs.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.2	UAF + U2F + FIDO2; >=L1	For each Authenticator Security Parameter, the vendor SHALL document the protections that are implemented for this parameter in order to support the FIDO Authenticator Security Goals or FIDO Authenticator Security Requirements, the location where this parameter is stored, how the parameter is protected in each storage location, how and when the parameter is input or output from the Authenticator, in what form the parameter is input or output, and when (if ever) the parameter is destroyed. Those Authenticator Security Parameters whose confidentiality MUST be protected in order to support the FIDO Security Goals or FIDO Authenticator Security Requirements SHALL be documented as “Secret Authenticator Security Parameters”; these SHALL, at minimum, include any of the following that are FIDO Relevant: secret and private keys, Allowed Random Number Generators’ state data, FIDO user verification reference data, and FIDO biometric data.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, the vendor SHALL describe the reliance of the Authenticator Application on the underlying operating environmentfor those Authenticator Security Parameters which are not fully maintained in the Authenticator Application.	Please refer to "ASP's Table" for the detailed documentation of all ASPs.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	

Key Management and Authenticator Security Parameters	2.1.3	UAF + U2F + FIDO2; >=L1	For each Authenticator Security Parameter that is a cryptographic key that is generated, used, or stored within the Authenticator, the vendor SHALL document how this key is generated, whether the key is unique to a particular Authenticator or shared between multiple Authenticators, and the key's claimed cryptographic strength. This claimed cryptographic strength SHALL NOT be larger than the maximal allowed claimed cryptographic strength for the underlying algorithm, as specified in the "Allowed Cryptography List" [FIDOAllowedCrypto]. If the key is used with an algorithm not listed on the "Allowed Cryptography List" [FIDOAllowedCrypto], then the claimed cryptographic strength for this key SHALL be zero. Note This requirement interacts with requirement 5.4 as the cryptographic strength of a key might get degraded - depending on potential side channel attacks - slightly each time the key is used.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, the vendor SHALL describe the reliance of the Authenticator Application on the underlying operating environmentfor those Authenticator Security Parameters (where stored, how protected, ...) which are not fully maintained in the Authenticator Application. If a cryptographic key is generated using an RNG with an unknown cryptographic strength, the cryptographic strength of that key is unknown.	Please refer to "ASP's Table" for the detailed documentation.	{A2} The tester SHALL verify that the documentation meets the requirement.	
Key Management and Authenticator Security Parameters	2.1.4	UAF + U2F + FIDO2; >=L1	The vendor SHALL document the Authenticator's Overall Claimed Cryptographic Strength; the Overall Authenticator Claimed Cryptographic Strength SHALL be less than or equal to the claimed cryptographic strength of all the Authenticator Security Parameters that are cryptographic keys. **Note** The security strength is a number associated with the amount of work (that is, the number of operations) that is required to break a cryptographic algorithm or system. It is specified in bits and it is often a value like 80, 112, 128, 192, 256.	L1: At L1, if the security strength for the RNG is not known, an unknown Overall Claimed Cryptographic Strength SHALL be assumed - which is allowed at L1.	Provide the Security Secretariat with a rationale of how the requirement above is met.	The overall cryptographic strength of the authenticator is 128.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.5	UAF + U2F + FIDO2; >=L1	All Authenticator Security Parameters within the Authenticator SHALL be protected against modification and substitution.	L1: At L1, the Authenticator Application SHALL follow best security practices specific to the underlying operating environment for protecting the Authenticator Security Parameters against being modified or substituted by (1) the	Provide the Security Secretariat with a rationale of how the requirement above is met.	ASPs stored within the authenticator are stored in the Secure Element, and thus rely on it for protection against modification and substitution.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	

Key Management and Authenticator Security Parameters	2.1.6	UAF + U2F + FIDO2; >=L1	All Secret Authenticator Security Parameters within the Authenticator shall be protected against unauthorized disclosure.	L1: At L1, the Authenticator Application SHALL follow best security practices specific to the underlying operating environment for protecting the Authenticator Security Parameters against being modified or substituted by (1) the	Provide the Security Secretariat with a rationale of how the requirement above is met.	ASPs stored within the authenticator are stored in the Secure Element, and thus rely on it for protection against unauthorized disclosure.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.7	UAF + U2F + FIDO2; >=L1	The Authenticator SHALL use an Allowed Data Authentication, Signature, or Key Protection Cryptographic Function to protect any externally-stored Authenticator Security Parameters against modification or the replay of stale (but possibly previously authenticated) data. **NOTE** In this requirement, externally-stored refers to parameters stored outside of the Authenticator boundary. For example, cloud storage services.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	externally stored ASPs are wrapped using AES128-GCM to protect externally stored data against replay of stale data.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.8	UAF + U2F + FIDO2; >=L1	The Authenticator SHALL protect any externally-stored Secret Authenticator Security Parameters using an Allowed Key Protection Cryptographic Function. [UAFAuthnrCommands], [Sections 5.1, 6.3.4] for RawKeyHandles.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	externally stored ASPs are wrapped using AES128-GCM. This is an allowed key protection cryptographic function	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.9	UAF + U2F + FIDO2; >=L1	Any key used with an Allowed Key Protection Cryptographic Function to protect an externally-stored secret or private key which is an Authenticator Security Parameter SHALL have a claimed cryptographic strength greater than or equal to the claimed cryptographic strength of the key being wrapped.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, externally-stored means stored outside the Authenticator boundary. In the case of L1 this Authenticator boundary includes the underlying operating environment.	The private key is ECDSA on P-256 (with strength of 128) and it is wrapped using AES128-GCM (strength of 128).	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.10	UAF + U2F + FIDO2; >=L1	Authenticators might offload the persistent storage of key material to components outside the Authenticator boundary if they cryptographically wrap it appropriately. Such structure containing cryptographically wrapped key material or information related to keys is called Key Handle containing a key (in [WebAuthn] the term Credential ID is used instead of Key Handle). If the Authenticator uses such Key Handle approach, the Authenticator SHALL verify that any Key Handle containing a key provided to the Authenticator was generated by that Authenticator using an Allowed Data Authentication or Signature Cryptographic Function; if not, then no signature using this key SHALL be generated. [U2FRawMsgs], [Section 5.1] and [UAFAuthnrCommands], [Annex A Security Guidelines, entry	L1: At L1, this Authenticator boundary includes the underlying operating environment.	Provide the Security Secretariat with a rationale of how the requirement above is met.	Our Authenticator implements the "keyhandle containing a key" approach. Whenever the authenticator receives a keyhandle from the RP, it verfies the authenticity of the keyhandle containing the key, using AES128-GCM before the key is used for any signing operation. Kindly see "signing operation" in the "DEVICE" sheet.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	

Key Management and Authenticator Security Parameters	2.1.11	UAF ; >=L1	If the Authenticator supports the KHAccessToken [UAFAuthnrCommands] method of binding keys to apps, then the Authenticator SHALL verify that the supplied KHAccessToken is associated with the referenced Key Handle prior to using that Key Handle to generate a signature; if not, then no signature associated with this Key Handle SHALL be generated. [UAFAuthnrCommands], [Section 6.3.4].	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	N/A this device does not support KHAccessToken method	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.12	UAF + U2F + FIDO2; >=L1	If the Authenticator supports the Key Handle approach, then the Authenticator SHALL verify that any Key Handle containing a key provided to the Authenticator is associated with the application parameter (U2F) or AppID (UAF) or RP ID (FIDO2) by using an Allowed Data Authentication or Signature Cryptographic Function; if not, then no signature using this key SHALL be generated. [U2FRawMsgs], [Section 5.1] and [UAFAuthnrCommands], [Section 6.3.4].	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	Our Authenticator implements the "keyhandle containing a key" approach. Whenever the authenticator receives a keyhandle from the RP, it verfies that the AppID is associated with the Key using AES128 GCM & SHA256. If both verifications did not pass then the key will not be used for any signing operation. Kindly see "signing operation" in the "DEVICE" sheet.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.13	UAF + U2F + FIDO2; >=L1	The Authenticator SHALL generate an independent User Authentication Key for each registration [UAFAuthnrCommands], [Section 6.2.4]. **Note** Any User Authentication Key (Uauth) SHALL only be used for authenticating one user account to one particular Relying Party.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	We generate an independent User Authentication Key for each generation	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.16	UAF + U2F + FIDO2; >=L1	In the event that an Authenticator Security Parameter is “destroyed” it is SHALL be made permanently unavailable so it can never be read or used again. **Note** The means by which this is accomplished is implementation and level dependent. It may by simply deleting it, overwriting it, destroying the key material used to encrypt it or other. **Note** The purpose of this requirement is primarily so that a factory reset carried out by an end user before they sell or dispose of their device giving assurance that the new owner cannot re instate authentication keys.	L1: At L1, the Authenticator Application SHALL follow best security practices specific to the underlying operating environment for protecting the Authenticator Security Parameters against being recovered and used.	Provide the Security Secretariat with a rationale of how the requirement above is met.	The Authenticator makes an ASP permanently unavailable by deleting the ASP data from the authenticator SE during a device reset.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.18	UAF + U2F + FIDO2; >=L1	Any time the Authenticator generates an Authenticator Security Parameter which is a key for use with an algorithm specified in the “Allowed Cryptography List” [FIDOAllowedCrypto], the Authenticator SHALL generate keys as required by the standard referenced in the “Allowed Cryptography List” [FIDOAllowedCrypto] for that algorithm.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	The authenticator generates keys based on the standsrd refrenced in the allowed cryptographic list (FIPS 140-2).	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	

Key Management and Authenticator Security Parameters	2.1.19	UAF + U2F + FIDO2; >=L1	Any wrapped FIDO biometric data and FIDO user verification reference data that is output from the Authenticator SHALL only be able to be unwrapped by the Authenticator that produced this data. **Note** Cryptographic Collision would be an exception.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	In this authenticator, the user verification data is not stored, rather it is hashed and the hash is wrapped using AES-GCM and the key is stored in the Secure Element which can only be accessed and used by the authenticator that produced it.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Key Management and Authenticator Security Parameters	2.1.20	UAF + U2F + FIDO2; >=L1	Any wrapped Authenticator User Private Key (UAuth.priv) that is output from the Authenticator SHALL only be able to be unwrapped by the Authenticator that produced this data.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	In this authenticator, the User Private Key is wrapped using AES-GCM and the key is stored in the Secure Element which can only be accessed and used by the authenticator that produced it.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Random Number Generation	2.2.1	UAF + U2F + FIDO2; >=L1	An Allowed Random Number Generator or Allowed Key Derivation Function SHALL be used for all key generation resulting in an Authenticator Security Parameter and for any random input for FIDO Relevant signature generation.	L1: At L1, the Authenticator Application SHOULD use the OSes RNG if it is an Allowed RNG according to [FIDOAllowedCrypto] and add entropy as described in [FIDOAllowedCrypto], section "Random Number Generator". Otherwise the Authenticator	Provide the Security Secretariat with a rationale of how the requirement above is met.	This authenticator uses a TRNG for all key generation.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
N/A	2.2.2	UAF + U2F + FIDO2; >=L1	The security strength (see the relevant Allowed Deterministic Random Number Generator specification document cited in the "Allowed Cryptography List" [FIDOAllowedCrypto]) of any Authenticator's Allowed Deterministic Random Number Generator SHALL be at least as large as the largest claimed cryptographic strength of any key generated or used. If the Authenticator generates a key with an Allowed Key Derivation Function, or uses a key with parameters generated by an Allowed Key Derivation Function (see the "Allowed Cryptography List" [FIDOAllowedCrypto]), then the security level of the Allowed Key Derivation Function SHALL be at least as large as the claimed cryptographic level of they key generated or used.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	N/A because This authenticator utilizes a TRNG	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	

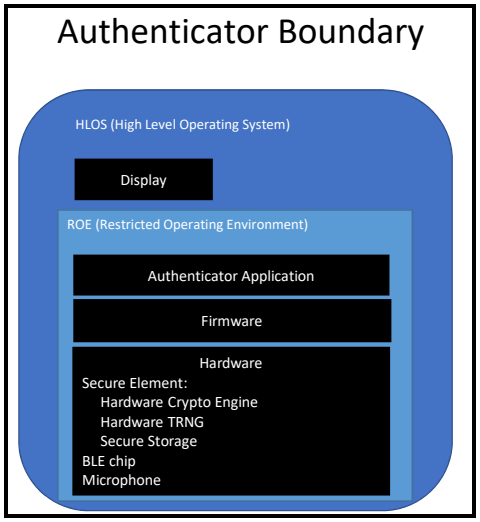
Random Number Generation	2.2.3	UAF + U2F + FIDO2; >=L1	If the Authenticator adds Authenticator generated nonces and the nonces are produced randomly, then an Allowed Random Number Generator SHALL be used for nonce generation. Authenticators with unrestricted keys (i.e. Metadata Statement isKeyRestricted: false) don't exclusively control the to-be-signed message and hence have no need to generate a nonce.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	The authenticator doesn't produce nonces.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Signature and Registration	2.3.1	UAF + U2F + FIDO2; >=L1	The vendor SHALL document whether the Authenticator supports Signature Counters and if they are supported, the vendor SHALL document whether one Signature Counter per authentication key is implemented or one (global) Signature Counter for all authentication keys (i.e. at least one counter covering multiple keys).	L1: At L1, Authenticators not running in an Allowed Restricted Operating Environment (AROE) [FIDORestrictedOperatingEnv], SHALL support signature counter(s).	Provide the Security Secretariat with a rationale of how the requirement above is met.	We support one (global) Signature Counter for all authentication keys.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Signature and Registration	2.3.2	UAF + U2F + FIDO2 ; >=L1	If the Authenticator claims supporting signature counter(s), then the Authenticator SHALL ensure that the signature counter value contained in FIDO signature assertions related to one specific authentication key either 1- is (a) greater than "0" and always has been greater than "0" for any previously generated FIDO signature assertion related to the same authentication key and is (b) greater than the signature counter value contained in any previously generated FIDO signature assertion related to the same authentication key, or 2- is set to "0" indicating that the signature counter is not supported any longer (e.g. in the case of a counter error). [U2FImplCons], [Section 2.6] and [UAFAuthnrCommands] [Section 6.3.4]. If one signature counter per authentication key is implemented (recommended option), it SHALL be incremented by 1 per signature operation. If a global signature counter is implemented, it SHOULD be incremented by a positive random number per signature operation (see [UAFAuthnrCommands] [Section A Security Guidelines, entry SignCounter]). **Note** Once a signature counter value contained in a FIDO signature assertion for one specific authentication key has been set to "0" in MUST stay at such value for that specific authentication key (due to the requirement 1).	No calibration required	Is this requirement applicable to the Authenticator? If No, then describe why. If Yes, provide the Security Secretariat with a rationale of how the requirement above is met.	Our authenticator supports a Global Signature Counter which is greater than '0' and is incremented by a positive random number per signature operation.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Authenticator's Test for User Presence and User Verification	3.1	UAF + U2F + FIDO2; >=L1	The Authenticator shall provide a mechanism to establish if the user authorizes a given action. (For a U2F, this is the “Test for User Presence”. Generically, the term “User Verification” may also refer to this “Test for User Presence”.) **NOTE** This requirement prevents remote attacks. The user has to confirm an action by pressing a button or providing some other gesture.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, in addition to the rationale provided by the vendor, this requirement MUST be demonstrated to the Test Proctor during Interoperability Testing. Documentation is not required.	This Authenticator uses Biometric Voice recognition to verify the user.	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	

Authenticator's Test for User Presence and User Verification	3.2	UAF + U2F + FIDO2; >=L1	The Authenticator is not marked as a Silent Authenticator [FIDOGlossary], the Authenticator SHALL NOT perform any authentication relevant operation without first establishing a user has requested the operation by verifying the user ([UAFAuthnrCommands], [section 6.2.4, 6.3.4]). An Authenticator without any keys for the specific user MAY allow the enrollment of new biometric reference data for that user without any additional user verification (bootstrapping user binding). Authentication relevant operations are: - Generating User Authentication Keys. - Producing signatures using such keys. - Adding any additional user verification methods. - Adding or changing user verification reference data sets (e.g. passwords or biometric templates). All such operations, with the exception of "Producing signatures using such keys" SHALL always require a fresh user verification (see requirement 3.4). With fresh user verification we mean a user verification that is performed at the time the respective operation to be approved by the user is triggered (and not before it)	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	The authenticator enforces user verification for all authentication relevant operations	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Authenticator's Test for User Presence and User Verification	3.4	UAF + U2F + FIDO2 ; >=L1	This requirement relates to "UserVerificationCaching" as specified in [UAFRegistry] for more details. If not declared otherwise in the Metadata Statement: Once the Authenticator's user verification / user presence check is successful, the user SHALL be deemed "verified" for no more than 10 seconds, or until the next operation which requires user verification, whichever comes first. Any provided User Verification Token SHALL NOT be valid after this time period. [UAFAuthnrCommands], [Appendix A Security Guidelines] If declared otherwise in the Metadata Statement: 1- The authenticator SHALL truthfully declare support of this user verification caching in the related Metadata Statement [FIDOMetadataStatement] (entry isFreshUserVerificationRequired=false). 2- Once the Authenticator's user verification / user presence check is successful, the user SHALL be deemed "verified" for no longer than the "maximum user verification caching time" as provided by the caller. If the caller has not specified a "maximum user verification caching time", then the Authenticator SHALL NOT cache the user verification event. Any provided User Verification Token SHALL NOT be valid after this time period. Multiple authentication operations might be performed in this time. The authenticator MAY limit the number of acceptable authentications in this time. 3- The authenticator SHALL add the "maximum user verification caching time" related to the specific Uauth key to the attestation statement. 4- When performing a TransactionConfirmation operation, the	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, in addition to the rationale provided by the vendor, this requirement MUST be demonstrated to the Test Proctor during Interoperability Testing. Documentation is not required.	A fresh user verification is required each time a user intends to do an operation. This authenticator does not support caching.	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	

Authenticator's Test for User Presence and User Verification	3.8	UAF + U2F + FIDO	The Authenticator SHALL protect against injection or replay of FIDO user verification data (e.g. user presence status, PIN, or biometric data).	L1: At L1, the Authenticator Application SHALL follow best security practices specific to the underlying operating environment for protecting against injection or replay of FIDO user verification data. This especially means that the Authenticator Application SHALL NOT	Provide the Security Secretariat with a rationale of how the requirement above is met.	The user verification data is SHA256 hashed and AESencrypted. The encryption keys are stored in the Secure element and therefore depends on the SE for protection against injection/replay	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Authenticator's Test for User Presence and User Verification	3.9	UAF + U2F + FIDO	Authenticators implementing user verification methods other than user presence check [FIDOGlossary], SHALL rate-limit user verification attempts in order to prevent brute force attacks. [FIDOMetadataStatement], sections 3.1, 3.2, 3.3 and [UAFAuthnrCommands], Appendix A Security Guidelines, entry "Matcher". The overarching requirement is based on an upper limit for the probability of a successful brute-force attack. The upper limits specified in "calibration" below. For the purposes of this requirement, a brute-force attack is defined as follows: The attacker tries all possible input combinations (e.g. passwords, PINs, patterns, biometrics...) in order to pass the user verification. In the case of biometric user verification, the attacker brings a potentially unlimited number of "friends" that can try whether their biometric characteristic is accepted (as false accept). In all cases the number of trials per time is limited by the verification speed of the authenticator and the integrity of the authenticator is not violated (e.g. no decapping of chips, no malware, ...) - since there are other requirements dealing with such attacks. **Note** - The rate limiting requirement applies to all user verification methods (other than user presence check) - Implementing a more strict rate limiting method is allowed. - We recommend 1- Allowing up to 3 failed user verification attempts without any penalty and then imposing a delay of at least 30 seconds before the 4th	L1: At L1, the time dependent probability of a successful brute-force attack on the authenticator SHALL be $P(t) \leq \text{maximum}(170/10000, (24*t+16)/10000)$, with t being the time in days. For a 4 digit PIN it means up to 170 non-biometric user verification	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, in addition to the rationale provided by the vendor, this requirement MUST be demonstrated to the Test Proctor during Interoperability Testing. Documentation is not required.	The authenticator implements user presence check, allowing up to 3 failed user verification attempts without any penalty and then imposing a delay of at least 30 seconds before the 4th one, increasing exponentially with each successive attempt (e.g., 1 minute before the 5th one, 2 minutes before the 6th one). After the 10th failed attempt, the device is reset to default mode and all resident keys are deleted.	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	

Privacy	4.1	UAF + U2F + FIDO2; >=L1	An Authenticator SHALL NOT have any Correlation Handle that is visible across multiple Relying Parties. If the authenticator uses a shared attestation key (e.g. Full Basic Attestation), the minimum number of Authenticators sharing this key MUST be at least 100000. **Note** The goal of this requirement is that, for privacy reasons, the Authenticator MUST NOT leak information about the user across multiple Relying Parties by sharing a Correlation Handle. This requirement specifically applies to KeyIDs/CredentialIDs, KeyHandles etc.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	The implementation of our keyhandle is output from the authenticator in encrypted form and hence does not reveal any correlation handle that is visible across multiple Relying Parties. we use shared attestation keys that are shared across more 100,000 authenticators.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Privacy	4.2	UAF + U2F + FIDO2; >=L1	An Authenticator SHALL NOT provide information to one Relying Party that can be used to uniquely identify that Authenticator instance to a different Relying Party.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	This Authenticator only provides to relying parties, the information that is relevant for the authentication (Signed challenge). There is no device-unique information that is provided to the RP.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Privacy	4.3	UAF + FIDO2; >=L1	An external party with two (AAID, KeyID) / (AAGUID, CredentialID) tuples produced using the Authenticator SHALL NOT be able to establish that they were produced using the same Authenticator.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met.	External party with two (AAID, KeyID) tuples produced using the Authenticator CANNOT establish that they were produced using the same Authenticator. The reason is because the keyhandle is encrypted & hence not readable and the AAID is shared by 100,000 other devices of thesame category.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Privacy	4.4	UAF ; >=L1	The Authenticator's response to a "Deregister" command SHALL NOT reveal whether the provided KeyID was registered.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, in addition to the rationale provided by the vendor, this requirement	The authenticator doesn't provide any information about the KeyID while responding to a "Deregister" command because it gives the same "OK" message anytime "deregister" command is triggered.	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	
Attestation	6.2	UAF + U2F + FIDO2; >=L1	Each Authenticator being declared as the same model (i.e. having the same AAID, AAGUID or having at least one common attestationCertificateKeyIdentifier in the MetadataStatement), SHALL fulfill at least the security characteristics stated for that Authenticator model.	No calibration required	Provide the Security Secretariat with a rationale of how the requirement above is met. At L1, in addition to the rationale provided by the vendor, this requirement MUST be demonstrated to the Test Proctor during Interoperability Testing. Documentation is not required.	All authenticators of this model fulfill thesame security characteristics which are declared for the model. This has been demonstrated during the interoperability test.	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	
Attestation	6.3	UAF + U2F + FIDO2; >=L1	The Authenticator SHALL accurately describe itself in its provided metadata, or alternately describe an Authenticator of lesser security. The vendor SHALL provide all mandatory Metadata Statement fields see [FIDOMetadataRequirements].	No calibration required	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	The Authenticator accurately describes itself in the metadata. This has been demonstrated during the interoperabiliy test.	{A0} The Security Secretariat SHALL verify the requirement during Interoperability Testing.	
Self-Tests and Firmware updates	8.2	UAF + U2F + FIDO2; >=L1	If the Authenticator mediates the update of its software, then the Authenticator SHALL use an Allowed Data Authentication or Signature Cryptographic Function, as required by the standard referenced in the "Allowed Cryptography List" [FIDOAllowedCrypto], to verify that the software being loaded has not been tampered with. If the loaded software does not pass, then the Authenticator SHALL NOT update the software.	No calibration required	Is this requirement applicable to the Authenticator? If No, then describe why. If Yes, provide the Security Secretariat with a rationale of how the requirement above is met.	The authenticator update files are SHA256-HMAC'd to protect them from tampering. The authenticator verifies the integrity of the updates software.	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	

Manufacturing and Development	9.1	UAF + U2F + FIDO2; >=L1	If Authenticator Security Parameters which are cryptographic keys are generated during manufacturing, then these keys SHALL be generated as required by the standard referenced in the “Allowed Cryptography List” [FIDOAllowedCrypto] for that algorithm using an Allowed Random Number Generator. .	L1: At L1, the creation of the final Authenticator Application is considered the Authenticator manufacturing.	Is this requirement applicable to the Authenticator? If No, then describe why. If Yes, provide the Security Secretariat with a rationale of how the requirement above is met.	The key which is generated during manufacturing is the device attestation key & it is generated according to the standard refrenced in the "Allowed cryptography list" (FIPS 140-2).	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	
Manufacturing and Development	9.4	UAF + U2F + FIDO2; >=L1	A revision control system SHALL be implemented for the Authenticator and all of its components, and for all associated Authenticator documentation. This revision control system SHALL, at minimum, track changes to all software or hardware specifications, implementation files, and all tool chains used in the production of the final Authenticator	L1: At L1, the use of a revision control system SHALL only be proven for the Authenticator Application.	Provide the Security Secretariat with a rationale of how the requirement above is met.	We use SVN as the revision control system to track all software & hardware specifications.	{A2} The tester SHALL conduct the documentation review described by the vendor, and confirm that all the results of this review meet the requirement.	
Manufacturing and Development	9.5	UAF + U2F + FIDO2; >=L1	Each version of each configuration item that comprises the Authenticator and associated documentation SHALL be assigned a unique identification. **Note** "Configuration item" stands for all the objects managed by the configuration management system during the product development. These may be either parts of the product (e.g. source code) or objects related to the development of the product like guidance documents, development tools, tests results, etc.)	L1: At L1, the configuration items comprising the Authenticator Application are relevant.	Provide the Security Secretariat with a rationale of how the requirement above is met.	We use SVN to manage assignment of unique identification for all configuration items that are used in the authenticator. (eg source code, unit test results, guidiance documents, etc)	{A1} The Security Secretariat SHALL review the provided rationale to verify the requirement is met.	



UVHash = SHA256 of (User Verification Template(UV))
RawkeyHandle = AES128 (SHA256(KeyID) + PrivKey)+ KeyID
Key Handle = AES128-GCM(RawKeyHandle + Counter + SHA256(AppID) + UVhash)

Signing Operation

1. Device receives the Keyhandle with a browser supplied AppID
2. Prompts user for voice verification & hashes the captured data.
3. Decrypts keyhandle AES-GCM
4. Verifies UVhash in the keyhandle with the hash of voice verification data in No.2
5. If user is verified, hash the received AppID and compare with the AppID hash that is inside the KeyHandle.
6. Validate counter
7. Access AES key for raw keyhandle using KeyID
8. Decrypt the encrypted part of KawkeyHandle
9. Verify the keyID hash
10. Sign

ASP Description

ASP	Data	Description	Secret ?	Strength	Where this is stored.	How this is protected.	How this is generated.	Input/Output	When this is destroyed.	Unique or Shared
PrivateKeys/ Uauth Private keys	ECDSA P-256 Curve	Private key used for signing operation	Yes	128	in the Keyhandle	wrapped and exported to RP in AES-GCM 128	Generated using TRNG in SE	during registration and authentication	N/A	Unique
User verification reference data	Biometric voice data hash	sample used for user verification before any operation	Yes	N/A	in the Keyhandle	wrapped and exported to RP in AES-GCM 128	during user verification	during registration and authentication	N/A	Unique
Key handle	array	Contains all information necessary to authenticate a user to the RP	No	128	with relying party	AES-GCM	during user registration	during registration and authentication	N/A	Unique
Device Attestation Key	Device Root key	Device Root key inserted at manufcturing	yes	128	in the SE	it relies on the SE for protection	during manufacturing	Never	Never	Shared with 100,000 other authenticators
Global Signature Counter	int 32	Keeps track of the signature done by the device	No	NA	in the Keyhandle	AES128	in the SE during device instantiation	during registration and authentication	Device Reset	shared for all keys on a device
RawKeyHandle	KeyID + PrivKey	concatenation of the AppID and PrivKey	Yes	128	in the keyhandle	AES128	during user registration	during registration and authentication	N/A	unique per user Registration
Secret Key	AES128 key	Symmetric key used to protect PrivKey	Yes	128	in the SE	it relies on the SE for protection	in the SE during user registration	Never	Device Reset	unique per user Registration
Device Key	AES128-GCM key	Symmetric key to protect Keyhandle	yes	128	in the SE	it relies on the SE for protection	in the SE during device instantiation	Never	Device Reset	Unique per authenticator device