Common Credential Types consist of (named 'Credential Profile'):	*Credential Profile*	*Credential Profile*	*Profile Configurator*	*Profile Configurator*	*Profile Configurator*	*Profile Configurator*	*Profile Configurator*	*Credential Profile*	*Credential Profile*	*Credential Profile*	*Credential Format*	*Credential Format*	*Credential Format*	*Credential Format*	*Credential Format*	*Credential Format*	*Credential Format*		
Format + Signature + Revocation + Key Mgmt Issuer + Key Mgmt Holder	**Credential Profile is commonly called**	**Credential Profile Description**	**Credential Format**	**Signing Algorithm**	**Revocation Algorithm**	**Key Management (Issuer)**	**Key Management (Holder)**	**Formal Specification**	**IPR Policy**	**Implementations**	**Selective Disclosure**	**Standardization (Body, Process)**	**Implementation Support (e.g. Libraries) / Active Community**	**Technology Readiness Level**	**Crypto Agility**	**Predicates**	**Rich Schemas/Semantic**		
Glossary >>>	*This is the term under which the profile is usually referred to in (technical) conversations, descriptions and discussions.*	*A narrative that describes the profile, ideally understandable for a layman.*	*Select Credential Format from dropdown list below. Data fields are automatically populated from detailed sub-tables below.*	*Select Signing Algorithm from dropdown list below. Data fields are automatically populated from detailed sub-tables below.*	*Select Revocation Algorithm from dropdown list below. Data fields are automatically populated from detailed sub-tables below.*	*Select Key Management (Issuer) from dropdown list below. Data fields are automatically populated from detailed sub-tables below.*	*Select Key Management (Holder) from dropdown list below. Data fields are automatically populated from detailed sub-tables below.*	*Does a formal specification for the complete tech stack exist?*	*If a Profile-specific IPR Policy exists, it need to be described here*	*Known implementations of the particular credential profile*	*Is the credential format capable of selective disclosure - presenting or revealing a subset of claims/attributes - without relying on architecture and protocol solutions like Just-in-Time issuance or a presentation transformation by a trusted third party?*	*Under which Standardization Body and which standards track/status is the credential format standardized?*	*How complex is the implementation? Which/How many useful software libraries available?*	*according to NASA-Scheme (http://www.artemisinnovation.com/images/TRL_White_Paper_2004-Edited.pdf)*	*Is the credential format capable to cover different cryptographic algorithms and achieve cryptographic agility, as demanded by regulations?*	*Is the credential Format able to produce general-purpose predicates, that means attestations without revealing the data, e.g. age over 18?*	*Is the credential Format able to communicate data or annotations that supports the semantic understanding of the credential data?*		
AnonCreds + CL + Indy Revocation + did:indy + link secrets	**AnonCreds**	Well-known default profile in all Hyperledger Indy implementations. A favoured profile due to selective disclosure and predicate capability as well as privacy-preserving revocation mechanism. Standardization in community spec in progress (July 2022).	**AnonCreds**	**CL**	**Indy Revocation**	**did:indy**	**link secrets**			Hyperledger Indy + Aries	yes (ZKP)	Community Spec (draft)	Hyperledger Indy & Aries	TRL 6 or 7	none	yes	no		
LD Proofs + BBS + Status List 2021 + did:web + credential as secret	**JSON-LD VCs with BBS (Credential is Secret)**	[profile description goes here]	**LD Proofs**	**BBS**	**Status List 2021**	**did:web**	**credential as secret**				depending on signature algorithm	W3C	tbd	TBD	yes	depending on signature algorithm	yes		
LD Proofs + BBS + Status List 2021 + did:web + did:key	**JSON-LD VCs with BBS (Holder DID)**	[profile description goes here]	**LD Proofs**	**BBS**	**Status List 2021**	**did:web**	**did:key**				depending on signature algorithm	W3C	tbd	TBD	yes	depending on signature algorithm	yes		
MDOC + ECDSA/EdDSA + + X.509 certificates + raw public keys (none jwk)	**ISO mDL**	[profile description goes here]	**MDOC**	**ECDSA/EdDSA**		**X.509 certificates**	**raw public keys (none jwk)**	iso 18013-5		Apple Wallet, UL, IDEMIA	yes (Hash&Salt)	ISO	tbd	TRL 7	yes	no	no		
VC-JWT + ECDSA/EdDSA + Status List 2021 + did:ion (long form) + did:ion (long form)	**JWT VC Presentation Profile**	https://identity.foundation/jwt-vc-presentation-profile/	**VC-JWT**	**ECDSA/EdDSA**	**Status List 2021**	**did:ion (long form)**	**did:ion (long form)**	https://identity.foundation/jwt-vc-presentation-profile/		MS Authenticator, Ping Wallet, Workday Wallet	no	W3C	tbd	TRL 8	yes	no	no		
ICAO DTC + ECDSA/EdDSA + SLTD database (travel and identity documents) + X.509 certificates + raw public keys (none jwk)	**ICAO DTC**	[profile description goes here]	**ICAO DTC**	**ECDSA/EdDSA**	**SLTD database (travel and identity documents)**	**X.509 certificates**	**raw public keys (none jwk)**				yes	ICAO	tbd	tbd	tbd	tbd	tbd		
x.509 + ECDSA/EdDSA + CRL + X.509 certificates + raw public keys (none jwk)	**x.509**	[profile description goes here]	**x.509**	**ECDSA/EdDSA**	**CRL**	**X.509 certificates**	**raw public keys (none jwk)**				no	ITUT	tbd	TRL 9	yes	no	no		
VC-SD-JWT + ECDSA/EdDSA + Status List 2021 + X.509 certificates + raw public keys (jwk)	**SD-JWT VCs (w/ X.509 for Issuers)**		**VC-SD-JWT**	**ECDSA/EdDSA**	**Status List 2021**	**X.509 certificates**	**raw public keys (jwk)**				yes (salted hashes)	IETF (OAuth WG)	tbd	TRL 3	yes	no	no		

Common Credential Types consist of (named 'Credential Profile'):	*Signing Algorithm*	*Signing Algorithm*	*Signing Algorithm*	*Signing Algorithm*	*Signing Algorithm*	*Signing Algorithm*	*Revocation Algorithm*	*Revocation Algorithm*	*Revocation Algorithm*	*Revocation Algorithm*	*Revocation Algorithm*	*Revocation Algorithm*	*Revocation Algorithm*	*Revocation Algorithm*	*Key Management (Issuer)*	*Key Management (Issuer)*	*Key Management (Holder)*	*Key Management (Holder)*	*Related to all sub-categories, combine with "Implementation Support"*
Format + Signature + Revocation + Key Mgmt Issuer + Key Mgmt Holder	**Standardization (Body, Process)**	**Recognition by government authorities (NIST, BSI, ...)**	**Implementation Support (e.g. Libraries) / Active Community**	**Technology Readiness Level**	**Hardware support**	**Unlinkability/Uncorrelatability**	**Standardization (Body, Process)**	**Recognition by government authorities (NIST, BSI, ...)**	**Implementation Support (e.g. Libraries) / Active Community**	**Technology Readiness Level**	**Observability**	**Tracability**	**Scalability**	**Performance**	**Infrastructure for Key Resolution**	**Key Rotation**	**Infrastructure for Key Resolution**	**Key Rotation**	**Implementation Support (e.g. Libraries)**
Glossary >>>	*Under which Standardization Body and which standards track/status is the signing algorithm standardized?*	*Is the signing algorithm recognized in regulatory frameworks of leading government bodies?*	*How complex is the implementation? Which/How many useful software libraries available?*	*What is the magnitude of production deployments? How many? Coverage?*	*Is the Signing Algorithm supported by common hardware-backed cryptographic implementations, such as Secure Elements, SecureEnclave, HSM, Strongbox, TEE, TPM*	*Is the Signing Algorithm capable of creating a credential scheme, such that two verification processes can not be linked/correlated by colluding Verifiers/Relying Parties*	*Under which Standardization Body and which standards track/status is the revocation algorithm standardized*	*Is the revocation algorithm recognized in regulatory frameworks of leading government bodies?*	*How complex is the implementation? Which/How many useful software libraries available?*	*What is the magnitude of production deployments? How many? Coverage?*	*Does the Verifier have the possibility to observe the revocation status beyond the presentation?*	*Does the issuer have possibilities to trace the usage of his issued credentials through the revocation mechanism?*	*At what scale has the algorithm/technology been demonstrated to work? Are there any known issues?*	*How performant is the revocation mechanism/for issuer, holder and verifier?*	*Is there any infrastructure required to resolve keys and/or validate identifier to key binding*	*Can the key refered to in a credential be replaced by another key?*	*Is there any infrastructure required to resolve keys and/or validate identifier to key binding*	*Can the key refered to in a credential be replaced by another key?*	*How complex is the implementation? Are useful software libraries available?*
Credential Profile vvv																			
AnonCreds + CL + Indy Revocation + did:indy + link secrets	none	not acknowledged (indendent crypto analysis published)	Hyperledger Ursa only	some	no	yes	AnonCreds	no	tbd	tbd	no	tbd	limitations by accumulator size	tbd	dlt	yes	none	no	*(auto generated)*
LD Proofs + BBS + Status List 2021 + did:web + credential as secret	DIF (intention to transfer to IRTF CFRG)	not acknowledged	tbd	tbd	no	yes	W3C	no	tbd	tbd	yes	no	tbd	tbd	web server	yes	none	no	*(auto generated)*
LD Proofs + BBS + Status List 2021 + did:web + did:key	DIF (intention to transfer to IRTF CFRG)	not acknowledged	tbd	tbd	no	no	W3C	no	tbd	tbd	yes	no	tbd	tbd	web server	yes	none	no	*(auto generated)*
MDOC + ECDSA/EdDSA + + X.509 certificates + raw public keys (none jwk)	ANSI	yes	many mature implementations	high	yes	no	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	none	no	none	no	*(auto generated)*
VC-JWT + ECDSA/EdDSA + Status List 2021 + did:ion (long form) + did:ion (long form)	ANSI	yes	many mature implementations	high	yes	no	W3C	no	tbd	tbd	yes	no	tbd	tbd	none	no	none	no	*(auto generated)*
ICAO DTC + ECDSA/EdDSA + SLTD database (travel and identity documents) + X.509 certificates + raw public keys (none jwk)	ANSI	yes	many mature implementations	high	yes	no									none	no	none	no	
x.509 + ECDSA/EdDSA + CRL + X.509 certificates + raw public keys (none jwk)	ANSI	yes	many mature implementations	high	yes	no	tbd	yes	tbd	tbd	yes	tbd	tbd	tbd	none	no	none	no	
VC-SD-JWT + ECDSA/EdDSA + Status List 2021 + X.509 certificates + raw public keys (jwk)	ANSI	yes	many mature implementations	high	yes	no	W3C	no	tbd	tbd	yes	no	tbd	tbd	none	no	none	no	

Credential Format	Selective Disclosure	Standardization (Body, Process)	Implementation Support (e.g. Libraries) / Active Community	Technology Readiness Level	Crypto Agility	Predicates	Rich Schemas/Semantic	Specification	IPR Policy
Glossary	*Is the credential format capable of selective disclosure - presenting or revealing a subset of claims/attributes - without relying on architecture and protocol solutions like Just-in-Time issuance or a presentation transformation by a trusted third party?*	*Under which Standardization Body and which standards track/status is the credential format standardized?*	*How complex is the implementation? Which/How many useful software libraries available?*	*according to NASA-Scheme (http://www.artemisinnovation.com/images/TRL_White_Paper_2004-Edited.pdf)*	*Is the credential format capable to cover different cryptographic algorithms and achieve cryptographic agility, as demanded by regulations?*	*Is the credential Format able to produce general-purpose predicates, that means attestations without revealing the data, e.g. age over 18?*	*Is the credential format able to communicate data or annotations that supports the semantic understanding of the credential data?*	*Where is the credential format specified?*	*What is the policy regarding intellectual properties associated with this technology?*
AnonCreds	yes (ZKP)	Community Spec (draft)	Hyperledger Indy	TRL 6 or 7	none	yes	no	https://anoncreds-wg.github.io/anoncreds-spec/	
LD Proofs	depending on signature algorithm	W3C	tbd	TBD	yes	depending on signature algo	yes		
MDOC	yes (Hash&Salt)	ISO	tbd	TRL 7	yes	no	no		
JWT	no	IETF	high	TRL 9	yes	no	no	https://datatracker.ietf.org/doc/rfc7519/	https://trustee.ietf.org/documents/trust-legal-provisions/
JWP	yes (ZKP, Hash&Salt)	DIF	tbd	TRL 3 or 4	yes	yes	tbd		
VC-JWT	no	W3C	tbd	TRL 8	yes	no	no		
SD-JWT	yes (salted hashes)	IETF (OAuth WG)	4 libraries under active development (Python, Kotlin, Rust, TypeScript)	TRL 4	yes	no	no	https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/	https://trustee.ietf.org/documents/trust-legal-provisions/
VC-SD-JWT	yes (salted hashes)	IETF (OAuth WG)	tbd	TRL 3	yes	no	no	https://datatracker.ietf.org/doc/draft-ietf-oauth-selective-disclosure-jwt/	https://trustee.ietf.org/documents/trust-legal-provisions/
x.509	no	ITUT	tbd	TRL 9	yes	no	no		
CWT	no	IETF	tbd	TRL 8	yes	no	tbd		
ACDC (KERI)	yes*	IETF (intention to go to Bloc	tbd	tbd	yes	possible	tbd		
CESR/CESR Proof	yes	IETF (intention to go to Bloc	tbd	tbd	yes	possible	tbd		
ICAO DTC	yes	ICAO	tbd	tbd	tbd	tbd	tbd		

Signing Algorithm	Standardization (Body, Process)	Recognition by government authorities (NIST, BSI, ...)	Implementation Support (e.g. Libraries) / Active Community	Technology Readiness Level	Hardware support	Unlinkability/Uncorrelatability	Specification	IPR Policy	Performance
Glossary	*Under which Standardization Body and which standards track/status is the signing algorithm standardized?*	*Is the signing algorithm recognized in regulatory frameworks of leading government bodies?*	*How complex is the implementation? Which/How many useful software libraries available?*	*What is the magnitude of production deployments? How many? Coverage?*	*Is the Signing Algorithm supported by common hardware-backed cryptographic implementations, such as Secure Elements, SecureEnclave, HSM, Strongbox, TEE, TPM*	*Is the Signing Algorithm capable of creating a credential scheme, such that two verification processes can not be linked/correlated by colluding Verifiers/Relying Parties*	*Where is the signature algorithm specified?*	*What is the policy regarding intellectual properties associated with this technology?*	
CL	none	not acknowledged (indender	Hyperledger Ursa only	some	no	yes	CL-Signatures of there own do not have a formal specification, that is included in Anoncreds		up to 7 seconds for a validation and approximately 30 seconds for credential definition generation
BBS	DIF (intention to transfer to IRTF C	not acknowledged	tbd	tbd	no	yes	https://datatracker.ietf.org/doc/draft-looker-cfrg-bbs-signatures/		
ECDSA/EdDSA	ANSI	yes	many mature implementation	high	yes	no	X9.62-2005		
RSA	tbd	yes, but support is fading	many mature implementation	tbd	yes	no			

Revocation Algorithm	Standardization (Body, Process)	Recognition by government authorities (NIST, BSI, ...)	Implementation Support (e.g. Libraries) / Active Community	Technology Readiness Level	Observability	Tracability	Scalability	Performance	Offline Friendliness	Specification	IPR Policy
Glossary	*Under which Standardization Body and which standards track/status is the revocation algorithm standardized*	*Is the revocation algorithm recognized in regulatory frameworks of leading government bodies?*	*How complex is the implementation? Which/How many useful software libraries available?*	*What is the magnitude of production deployments? How many? Coverage?*	*Does the Verifier have the possibility to observe the revocation status beyond the presentation?*	*Does the issuer have possibilities to trace the usage of his issued credentials through the revocation mechanism?*	*At what scale has the algorithm/technology been demonstrated to work? Are there any known issues?*	*How performant is the revocation mechanism/for issuer, holder and verifier)?*	*Does the revocation mechanism support an authentication if the holder is offline?*	*Where is the revocation mechanism specified?*	*What is the policy regarding intellectual properties associated with this technology?*
Status List 2021	W3C									https://w3c-ccg.github.io/vc-status-list-2021/	
CRL	tbd	yes	tbd	tbd	yes	tbd	tbd	tbd	tbd		
Indy Revocation	AnonCreds	no	tbd	tbd	no	tbd	limitations by accumulator si	tbd	tbd		
BBF18- cryptographic accumulator based on RSA	tbd	no	tbd	tbd	no	tbd	tbd	tbd	tbd		
VB20 - cryptographic accumulator based on pairings	tbd	no	tbd	tbd	no	tbd	tbd	tbd	tbd		
Non-Revocation Token	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd		
SLTD database (travel and identity documents)									tbd		
Validity Credential	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd	tbd		

Key Management	Infrastructure for Key Resolution	Key Rotation	Key History	Implementation Support (e.g. Libraries) / Active Community	Specification	IPR Policy
Glossary	*Is there any infrastructure required to resolve keys and/or validate identifier to key binding*	*Can the key refered to in a credential be replaced by another key?*	*Is it possible to retain and obtain the history of keys related to a certain identifier?*	*How complex is the implementation? Which/How many useful software libraries available?*	*Where is the key management specified?*	*What is the policy regarding intellectual properties associated with this technology?*
raw public keys (jwk)	none	no	no		https://www.rfc-editor.org/rfc/rfc7517	https://trustee.ietf.org/documents/trust-legal-provisions/
link secrets	none	no	no			
X.509 certificates	none	no	no			
did:indy	dlt	yes	yes			
did:ion (long form)	none	no	no			
did:ion (short form)	yes	yes	yes			
did:key	none	no	no			
did:peer			?			
did:ebsi	dlt	yes	yes(?)			
KERI	witness network	yes	yes			
did:web	web server	yes	no			
raw public keys (none jwk)	none	no	no			
credential as secret	none	no	no			
jwks_uri	web server	yes	no			