Executive Summary

July 13, 2020

This document is part of an ongoing series of informal discussions with the European Commission's Directorate-General for Communications Networks, Content and Technology (DG CONNECT), the European Union Agency for Cybersecurity (ENISA), the European Telecommunications Standards Institute (ETSI), and browser vendors such as Apple, Google, Microsoft, Mozilla, Opera, and Vivaldi.

This document collects concerns from Apple, Google, Microsoft, Mozilla, Opera, and Vivaldi around a recently proposed use of Attribute Certificates (RFC 5755) to communicate Qualified Website Authentication Certificates (QWACs), established by EU Regulation 910/2014 (the eIDAS Regulation). Browser vendors are concerned that, as proposed, the use of Attribute Certificates to communicate QWACs (hereafter called "ac-QWACs") will encounter significant interoperability concerns with existing software and which are highly likely to damage and harm the security, privacy, and interoperability functions of these browser vendors' products. An alternative proposal for QWACs was previously put forward by browsers, called nt-QWACs, that would entirely avoid these concerns, and thus remove barriers to the adoption and use of QWACs within the Digital Single Market.

The concerns with ac-QWACs specifically, but which also generalize to other forms of Trust Service Provider (TSP) mediated QWACs, such as the existing ETSI EN 319 412-1/412-2 approaches, are that such solutions are:

i. Incompatible with how browsers load web pages and resources; and
ii. Incompatible with how websites use and deploy TLS; and
iii. Incompatible with how users browse the web; and
iv. Not technologically neutral with current and future methods of establishing TLS; and
v. Problematic with respect to user privacy; and
vi. Inconsistent with the text of the Regulation and the stated requirements and goals.

These concerns are expanded on in greater detail within this document. The design of nt-QWACs, as proposed by browsers, was specifically designed with these considerations in mind, and on the basis of the stated purpose, goals, and requirements within the eIDAS Regulation. Solutions such as ac-QWACs, as proposed, or tls-QWACs, as currently specified by ETSI, represent significant barriers to adoption, interoperability, and use, while also greatly jeopardizing the security of these products and their users, by unnecessarily restricting the agility and ability to respond to security threats and evolving security improvements.

Background

Throughout the latter half of 2019, and continuing through 2020, representatives of major browser manufacturers, including those of Apple, Google, Microsoft, Mozilla, Opera, and Vivaldi, worked to establish a technical proposal regarding Qualified Website Authentication Certificates (QWACs). This proposal was aimed at addressing the concerns shared by DG CONNECT and ENISA regarding the limited uptake and interoperability of QWACs within these vendors' respective products and trust frameworks. Building on the technological neutrality and interoperability goals set forward by the eIDAS Regulation, this proposal explored how a small adjustment to the specific technical profile of QWACs proposed by the ETSI ESI Technical Committee could be made easier to use, obtain, and deploy by websites.

In March 2020, browser manufacturers provided a demonstration on how an unmodified browser could consume and use such QWACs, called "nt-QWACs" to highlight their subtle distinction from the existing ETSI ESI specification, and without relying on any third-party services. During this meeting, ETSI ESI raised a concern that, despite the technological neutrality of the nt-QWAC solution, it was perceived as inadequate for security, due to only cryptographically binding organizational and legal identity to a website, and not to a specific protocol, such as TLS, or to a specific communication channel using TLS. Browser manufacturers expressed concern that ETSI ESI's interpretation did not seem consistent with Annex IV of the Regulation, nor would binding QWACs to TLS facilitate the goals of interoperability and technological neutrality. As a follow-up item, ETSI ESI was to provide a security analysis that explored the security properties needed of QWACs, in order to better identify solutions that could meet those properties.

In a follow-up meeting of June 2020, ETSI ESI shared their position, which was that a protocol-specific, technology-specific cryptographic binding was necessary under the Regulation, but without stating the specific and necessary security requirements or goals to be achieved. In addition, a demonstration based upon Attribute Certificates (RFC 5755) was provided to browsers. This demonstration showed an Attribute Certificate, or ac-QWAC, that included a binding to a specific TLS server certificate as one of the attributes. In order to validate this certificate, the user was directed to a third-party service, which would validate the ac-QWAC against the Trusted Services List, and then attempt to validate the cryptographic binding from the perspective of this third-party service. The subsequent discussion following this demonstration largely focused on concerns raised by browsers with respect to the cryptographic binding of the ac-QWAC to the TLS certificate. The particular technique used to establish this binding, which was represented as necessary by ETSI ESI, was seen by the browsers as creating conflict with many of the objectives of technological neutrality, interoperability, ease of deployment, ease of operation, privacy protection, and with the Regulation itself. As a follow-up deliverable, browser vendors agreed to provide an explanation of these concerns, detailing how

TLS is used within their products, why the proposed ac-QWAC has significant limitations not shared by nt-QWACs, and suggestions for how to address these concerns.

Incompatible with how browsers load web pages and resources

A common misconception with how browsers work is that the act of loading a URL, such as "https://google.com", is guaranteed to establish one and exactly one connection to that host. When using TLS, as HTTPS does, it's assumed that because there is a single connection, any properties of that TLS connection can be assumed for the "website" at the URL. Unfortunately, web browsers have not behaved like this since the very earliest days of the web, having changed even before browsers like Netscape Navigator and Microsoft Internet Explorer were even available.

A less common, but equally problematic assumption, is believing that even if there are multiple connections to a host, such as for "https://google.com", they will all share the same properties.

When a modern web browser loads a URL such as "https://google.com", it goes through a complex, sometimes vendor-specific, state machine. For example, the browser may examine a cache of resources on disk or in memory. If it has a resource that was previously retrieved, and still valid, it will load that resource, and its associated properties, from disk. The website is generally in control of how long these resources are cached, and may have configured them to be valid for hours, minutes, years at a time, and may even configure them to never expire or be invalidated.

If a resource on disk isn't available, the browser may attempt to establish a connection to the server. If it was previously connected to the server, it may have one or more connections existing within a "connection pool", a cache of connections. These may have been established mere seconds ago, or they might be long-lived, from minutes or even hours ago, depending on the browser and the users' activity. Each of these connections can be seen as fully independent: there is no guarantee that all connections in a pool will share the same properties, such as being to the same IP address or share the same certificate.

If the browser needs to establish a new connection, it will typically attempt to resolve the host via DNS. The browser may make only a single DNS request, or it may make multiple requests in parallel, for different record types. As IP addresses are returned, it may begin establishing connections in parallel, "racing" them to see which connection will be fastest. The winner is the connection that is used, but the other connections, to different IP addresses and potentially using different protocols entirely, such as HTTP/2 or HTTP/3, are often returned to the connection pool for subsequent use later.

Once the browser has established a connection, it will attempt the initial request. If, as in the case of "http://google.com", the browser receives a redirect from HTTP to HTTPS, it may repeat this whole process again to a new host, establishing a new connection with new properties. However, even if the redirect is to the same host, but a different URL, there's no guarantee that the connection used for the second URL will be the same as the first. This is because browsers, after loading a URL, typically return a connection back to the connection pool. When the second URL, for the same host, is loaded, there is no guarantee that it will get the same connection as the first.

As the web page loads, the browser will begin to load subresources of the page. These are the elements, such as images or stylesheets, that affect how the web page is displayed to the user. Until certain subresources are loaded, the webpage often appears "blank", and may simply display an empty white page. Each of these subresources is referenced by URLs, and so have all the same complexity as described above, repeated in a microcosm for each of the subresources. Each of these subresources may be on different hosts, and each of these hosts may have different certificates, and may be operated by different entities: content delivery networks, authentication/single-sign-on providers, analytics, Javascript libraries or frameworks, or any other number of partners that the website may partner with or delegate to. These subresources may have been cached, and so have older certificates than the original URL, or if the original URL was loaded from the cache, the subresources may have newer certificates.

If the original page makes use of a technology called inline frames, it's possible to load entire web pages within a web page, repeating all of these previously mentioned steps for loading, and without any distinct URL bar or other user-facing display. The use of iframes is a common pattern for enabling single sign-on or payments, including enabling compliance with the Revised Payment Services Directive (PSD2), Directive (EU) 2015/2366. The use and appeal of inline frames is precisely because they integrate with the "hosting" web page transparently, appearing as if they were just another subresource, despite being operated by a third-party service provider.

Modern technologies, such as WebSockets, Service Workers, Alt-Svc, and the HTTP/2 ORIGIN frame, can all further affect and compound the complexity here, by altering what host or connection a resource is loaded from, depending on when and how it's loaded within a page.[1] Similarly, modern web applications are able to make use of a wealth of persistence APIs, such as cookies, Web Storage, IndexedDB, and the Cache API, that allow the page itself to store data in a prevision session and load it within a future session. As with resource loading, these data storage mechanisms rely upon the concept of the Origin, and without any consideration of the certificate and/or connection properties.

[1] For further information, many of these concepts are discussed in the 2013 book "High Performance Browser Networking" by Ilya Grigorik, published by O'Reilly Media Inc. This book is available under a Creative Commons license at https://hpbn.co/ , and in particular, https://hpbn.co/primer-on-browser-networking/ discusses the disconnect between resources used for websites and sockets.

All of this can happen before a single pixel is displayed to the user or the webpage is said to be loaded. It is this complexity, which is necessary to support a rich Web application platform, that has consistently resulted in browsers largely rejecting web technologies that attempt to build on properties of the connection, because there is no single “the” connection to build on. The certificate is an example of a property of a connection, as there is no guarantee that the certificate used on one connection will be the same as on another connection, and that’s explicitly not a goal that browsers have wanted to support or have developers assume.

The reason all of these technologies work, and web browsers are able to continue to innovate and support rich, interoperable, interactive web applications, is that all of the Web’s security technologies are not based on sockets, connections, or key pairs, and instead built on a concept known as the Origin. The Origin is how web browsers establish security boundaries, and knows the difference between “this” webpage and “that” webpage.

Overly simplified, for most URLs, the Origin is made up of three properties: the scheme (e.g. “http” vs “https”), the host (e.g. “google.com”), and the port (such as 443, the default port for the scheme “https”). If a resource is “same-origin” to the page, it’s considered as part of the same security boundary, while if a resource is “cross-origin”, it’s treated as part of a separate security boundary, and care is taken when loading. Despite the myriad of connections, hosts, and certificates that were previously described, it is their Origin, which can generally be extracted directly from their URL, that affects the page security.

When browsers say that “only the host matters” in a certificate, not the key within the certificate, they are reflecting the fact that all of the webpage’s security and privacy properties are computed based on the Origin, which only takes the hostname into consideration. As long as an individual connection presents a certificate for that hostname, and a successful TLS handshake occurs, any other properties of the certificate do not affect the security of the webpage.

Both tls-QWACs and ac-QWACs rely on the ability to bind a resource load to an individual TLS connection and to a specific certificate, neither of which may reflect the current state of the web server, due to the loading process mentioned previously. As such, this binding is fundamentally incompatible with the actual loading mechanisms. As properties of connections, such as certificates, do not affect the computation of the origin, the existence of a tls-QWAC or an ac-QWAC does not affect the security or processing of the webpage and its Origin.

Similar technologies, like Extended Validation Certificates, are equally limited. The fact that Extended Validation certificates do not affect the Origin has been why some browsers have highlighted that such certificates do not meaningfully affect the security of a webpage. Due to how browsers work and load resources, only things that affect the Origin can be said to definitively and consistently affect the security of a webpage. Due to being bound to individual connections, rather than affecting the Origin, tls-QWACs and ac-QWACs can thus be said to suffer from the same issues noted by the security community that resulted from the introduction of Extended Validation certificates.

Changing how the Origin is computed is not a viable path forward, as it would involve fundamentally rearchitecting how web browsers and web security works, in a backwards-incompatible manner that would likely take decades. Incorporating these details into the Origin is also non-trivial and unlikely to succeed. This is because the Origin needs to be able to be determined in a connection-independent manner, using only the URL of the resource and the context that the URL is requested (e.g. the page that's requesting it, the type of HTML tag being used), as the Origin affects how connections are established, and thus needs to be known independent of the connection itself. This means integrating any changes to the Origin would require introducing entirely new URLs to the webpage, and potentially replacing DNS altogether.

While there have been, at various times, proposals to attempt to surface certificate information to web pages, e.g. via the Fetch API or the Resource Timing API, such proposals have been on the basis of diagnostic logging functionality, because they cannot and do not affect the security of a website itself, as they do not affect the Origin. Similarly, such APIs would expose a non-trivial amount of users to significant privacy risks, such as persistent tracking by websites, and thus even for diagnostic purposes represents a significant concern.

Absent such a significant and fundamental re-architecture of the Origin, which would redefine and redesign all Web technologies, any connection-dependent relationship, such as binding a QWAC to a certificate or TLS connection, provides no meaningful security to end users within any of the existing web browsers.

Incompatible with how websites use and deploy TLS

Solutions such as ac-QWACs, or any other form of cryptographic attestation in which the TSP makes the attestation, as suggested by ETSI ESI, are also incompatible with how modern websites are used and deployed, because as currently proposed, they do not account for the complexity present in the web today. As a consequence, such solutions would face significant, potentially insurmountable, barriers to widespread adoption.

Modern websites may, at any given point in time, have dozens of certificates in use. Coordinating global deployment across datacenters and points of presence is quite complex, and so at any given moment, users may encounter a myriad of certificates, potentially even from different CAs entirely, depending on what version of configuration a given server is running. For example, at the time of writing this, Google has fifty certificates in use just for "`www.google.com`", each with their own key.

While this example only applies to a single domain name, this effect is amplified when considering the Same Origin Policy, as discussed with how browsers operate. Websites regularly establish new domains and subdomains, which help them establish different security principals, much like modern operating systems use different user accounts and privileges. For example, a site might have domain names like "`website.example`", a sub-domain “`img.website.example`” hosted by Content Delivery Network (CDN) 1, a sub-domain “`fonts.website.example`” hosted by a different CDN, and may introduce new subdomains like “`user1.website-users.example`”. As best practice is to discourage wildcard certificates, each of these domains will have their own certificate, each with their own certificate management lifecycle, but all logically part of the same legal or natural identity.

While a solution might appear to allow expressing multiple certificates, it means that a website deploying QWACs such as tls-QWACs or ac-QWACs would need to consult with and receive approval from their TSP each time they wished to change or update their configuration. This is because it would require obtaining a new tls-QWAC or ac-QWAC any time the website wished to change or manage their certificates. This would be quite challenging, given that ETSI ESI’s specifications prohibit QWACs from being approved automatically, and require two independent approvers from the TSP to manually approve such issuance, due to ETSI ESI normatively incorporating the EV Guidelines. This would greatly impair modern best practices, like agility, but could also greatly hinder how a website is able to respond to a security incident, by requiring the TSP to approve before they implement a security fix to protect users.

For sites that do not have such global presences like Google, they may choose to let their CDN or hosting provider manage TLS for them. These providers automatically handle issuing and maintaining certificates for the website, making it easier and hassle-free to enable TLS. For such websites, they do not manage certificates themselves, so they do not know when or how their certificates will be rotated. While they are in full control of their domain, and their content, their service provider handles certificates for them. tls-QWACs and ac-QWACs are incompatible with such deployments, because they require that the website operator directly manage certificates themselves, at greater cost and complexity to their organization.

This constant changing of certificates emphasizes the importance that any cryptographic binding should ideally be oriented around the domain name, which is immune from such changes, rather than to the certificate or TLS. If a cryptographic binding to certificates is pursued, then as discussed in greater detail further within this document, such a binding needs to be manageable by the website itself, rather than the TSP. This ensures that website operators are able to manage and deploy certificates as necessary, enabling them to respond to security needs without delay, and allowing them to pursue more secure deployment options than purely self-managed web hosting. Similarly, any binding to a certificate fundamentally needs to consider multiple bindings and certificates, which change frequently and sometimes on-demand/instantaneously.

Incompatible with how users browse the Web

It’s widely understood that, for a non-trivial portion of the Web, users’ browsing activity is intercepted and scanned as part of normal operation. For users on home computers, this is often through one or more antivirus programs, which perform TLS interception and modification, in order to protect users from viruses and other threats the vendor may determine. For users on corporate computers, this may be done by a local enterprise administrator in order to provide similar traffic inspection and management capabilities to the organization. A common explanation by businesses is that they use such capability to detect and prevent malware and other forms of data-loss from exfiltrating their network.

While such products can introduce a host of security concerns on their own, they are still widely used and deployed. After the IETF rejected formalizing such protocols, due to the security harm they pose, the ETSI CYBER TC took such work on, formalizing protocols such as the Middlebox Security Protocol used to intercept and manipulate network traffic. While unfortunate for security, this demonstrates the perceived need within some industries for the ability to perform such traffic inspection and manipulation.

Unfortunately, cryptographic bindings to certificates or TLS are fundamentally at odds with such solutions, as they rely on altering the cryptographic properties of the connection. Solutions such as tls-QWACs or ac-QWACs are fundamentally incompatible with the vast majority of home and enterprise antivirus solutions, due to their integrated TLS interception capabilities, which break such certificates. While it may be possible to devise a protocol to exchange this information, between the middlebox and the client browser, such solutions are measured by years of standardization and adoption by vendors, followed by more years of adoption by users slowly upgrading to products that might support this hypothetical protocol. In the interim, websites would be unable to leverage the added validation provided by ac-QWACs or tls-QWACs.

By contrast, the design of nt-QWACs considered this use case as part of the rationale for limiting the cryptographic binding to the origin. By emphasizing the authentication of the domain and the entity operating the domain, the design enables users to be in control of their network activity, inspect it or otherwise enforce security requirements, such as on-the-fly phishing protection, while still being confident about the entity operating the domain.

Not technologically neutral

A significant challenge with ac-QWACs, as well as that of tls-QWACs, is that they are not technologically neutral in how they provide information about the operator of a website. Rather than providing information about the genuine and legitimate entity of a website, as represented by an Origin, the current implementation of tls-QWACs and the proposed ac-QWACs would be limited to only providing that information if the TLS protocol is used. This limits their ability to work with existing technologies, including alternative uses of TLS, as well as limits their ability to evolve and adapt to new technologies that may replace TLS. It also prevents a broader adoption of QWACs from being adopted in environments that could benefit from the added information, but do not use TLS.

An example of an existing technology that tls-QWACs impede on, and which ac-QWACs are incompatible with, is that of DANE TLSA. DANE TLSA can be used to securely establish a TLS connection to a website without the use of certificates at all. Unfortunately, tls-QWACs prohibit this method of connecting to a domain, because they necessitate the use of certificate-based key agreement, and ac-QWACs are incompatible with it. Ideally, QWACs would be agnostic about whether or not the TLS connection itself uses certificates to establish acceptable symmetric keys, and instead focus on providing information about the domain itself.

As proposed, both ac-QWACs and tls-QWACs only work with protocols involving TLS. This greatly limits their ability to be used in other protocols where such information would be useful, such as SMTP. Although SMTP can use TLS, SMTP may traverse multiple Mail Transfer Agents (MTAs) before making it to the actual recipient's mail server. As a result, forms of TLS authentication, such as TLS mutual authentication, are incapable of being used; or, at least, not without restricting SMTP to unscalable direct communication between the sender MTA and recipient MTA.

Rather than use TLS mutual authentication, SMTP servers frequently use DomainKeys Identified Mail (DKIM), which is based within DNS itself, similar to DANE. DKIM is used to authenticate and validate the original sender domain of an email, and can be used in hop-by-hop validation and by the final recipient MTA. Solutions like nt-QWACs are able to interoperate with such technologies, because they are associated with domains, like DKIM, rather than certificates. This technological neutrality can enable greater confidence in email, and reduce spam and misclassification, by allowing “sender reputation” to be associated with legal entities behind the domain names, rather than just the domain names as often done today. Such use cases cannot be met with tls-QWACs or ac-QWACs without significant changes to how SMTP servers operate, in ways that would greatly impair global email delivery. However, nt-QWACs, by being technologically neutral, are able to easily integrate with technologies like SMTP and DKIM, providing greater confidence and more reliable mail delivery.

Problematic to User Privacy

While the privacy issues with the use of eSignature certificates on the Web are well documented, solutions such as ac-QWACs also pose their own form of privacy risk. As discussed in the context of browser incompatibility, browsers intentionally separate out the relationship between sockets/connections and resources/websites. This fundamental architecture split makes it difficult or impossible for a website to determine information about the connection the user used to fetch one or more resources, and thus, makes it impossible to validate any per-connection/per-certificate cryptographic binding that might be present. While the browser incompatibility discussion explores why it is a fundamental architectural necessity to avoid exposing connection-specific information, this incompatibility also leads to privacy issues for users if ac-QWACs are deployed.

As presented as part of the Informal Workshop in June 2020, in order to validate ac-QWACs, a website must reveal a user's browsing activity to a third-party validation service, in order for that third-party validation service to fetch and validate the ac-QWAC, as well as connect to the server and validate the binding. Beyond this external validation serving no security purpose, as there is no way to guarantee that the third-party service's view of the server's certificate is aligned with any properties of any of the user's connections, past or present, it also represents a significant privacy issue.

This privacy issue is not one that can be solved purely by policy alone. For example, even if the third-party were to commit to not record or retain any records of users' browsing activity, the mere act of transmitting this information is largely indistinguishable from a privacy-problematic advertising technique known as "URL decoration" or "link decoration". As browsers continue to compete in areas of user privacy, steps are being taken to detect and attempt to prohibit "link decoration" from functioning at all.

Solutions like ac-QWACs, due to their per-connection/per-certificate cryptographic binding, inherently rely on this, and thus may soon find themselves incompatible with modern web browsers. However, solutions that limit the cryptographic bindings to the Origin, such as nt-QWACs, can still function. This is because nt-QWACs supports the delivery of all of the necessary validation information (e.g. OCSP responses, trust lists, etc) by the Origin itself, avoiding the need to involve any third-parties at all or reveal any browsing activity of the user, and without any modifications or changes to the privacy or security behaviors of modern browsers.

Unfortunately, the necessity for third-party involvement is also the result of protections browsers take with respect to user privacy. It is not possible to verify such a binding in JavaScript itself, because information about a connection is not exposed to a web page in order to protect user privacy. Although deprecated technologies like Flash previously allowed such inspection, which enabled useful academic research and attempts to classify security threats, it also provided a ready channel for fingerprinting and identifying users, and is now actively prohibited. Further, as discussed in how browsers work, exposing certificate information does not match how browsers manage and expose requests and sockets to web pages.

Such information is, for the most part, also not available to extensions to the browser that the user may install. Although at least one browser provides this information, other browsers have so far declined to do so, on the basis that exposing such APIs may represent more patterns for abuse than legitimate use, and undermine the security, agility, and privacy of users.

As a consequence, there is no interoperable way, either within a webpage or within a web browser extension, to access and/or validate any per-connection/per-certificate cryptographic binding that may exist. These choices are guided by emphasizing user privacy, including keeping local network information private from servers, which can use such information to identify or track users, or to coerce them into otherwise problematic practices, such as detecting and disabling anti-virus.

Inconsistent with the Regulation

Whether or not an Origin-based cryptographic binding is sufficient ultimately depends on the problem that QWACs are trying to address. As has been suggested by members of ETSI ESI, a cryptographic binding to the connection or protocol is viewed as necessary to fulfill the obligations of the eIDAS Regulation. However, that conclusion seems at odds with the text of the Regulation. Combined with the myriad concerns above that demonstrate how such cryptographic bindings fail to provide any additional assurance or utility to websites, pursuit of a cryptographic binding to a connection or TLS certificate presents a seemingly unnecessary complication that only serves to make deploying QWACs harder.

With respect to the objectives of QWACs, Recital 64 of the Regulation sets forth the objective of QWACs as being that "by which a visitor to a website can be assured that there is a genuine and legitimate entity standing behind the website." Further, Recital 64 notes that, among other things, the Regulation "should not impede the use of other means or methods to authenticate a website not falling under this Regulation". In the context of Recital 64, authentication of a website is a means of associating a natural or legal entity with a website. As noted in the discussion of how browser technology works, a website is defined by its Origin; that is, the scheme, host, and port.

The precise requirements for website authentication certificates, that is, those that associate a legal entity with a scheme, host, and optional port, are laid forth in Annex IV of the Regulation. Compared to Annex I and Annex III, Annex IV does not necessitate the inclusion of electronic signature validation data. Electronic signature validation data is the means by which a message sent to a relying party, such as a signed document or, in the case of TLS, a CertificateVerify protocol message, can be tied back to the overall certificate. Annex IV only requires the inclusion of the domain name(s) operated by the natural or legal person.

In this regard, the Parliament and Council have shown excellent foresight in ensuring the technological neutrality of QWACs, and ensured that all of the information necessary and sufficient to authenticate a website (i.e. a domain name) to a legal or natural entity is present. It was to this end that nt-QWACs were proposed, based on a fulfillment of the Annex IV requirements as written, and in line with how web browsers and certificate consumers function.

Alternative Cryptographic Binding Approaches

Although these many issues are systemic, and thus any form of cryptographic binding to certificates significantly reduces the opportunity for adoption of QWACs, there are still better alternatives than the proposal set forth by ac-QWACs.

Of the many limitations of ac-QWACs, the biggest is that it requires the TSP to mediate every cryptographic binding to a certificate, issuing a new ac-QWAC whenever a TLS certificate or key changes changes. As this design requires the TSP issue a new ac-QWAC each time the TLS information changes, it’s equivalent to requiring TSPs to issue a new certificate each time a Subscriber wants to sign a document or e-mail. As discussed in the context of how modern and secure websites are administered, the ideal process is one highly automated and transparent to the website operator, which is the opposite of what ac-QWACs propose.

Solutions like electronic seals and signatures work by granting a Subscriber a signing certificate, and allowing them to sign whatever messages or documents they wish to sign, without needing to involve the TSP once the initial certificate has been issued. The same design is applicable here: rather than having the TSP perform the cryptographic binding between the QWAC and the TLS certificate, allow the Subscriber to establish the binding, the same as they do when signing documents or e-mail.

This design is a significant improvement over the existing ac-QWACs, because it removes the need to involve a third-party, the TSP, when managing websites, domains, and business relationships like CDNs, all of which can affect the certificates involved and thus necessitate new ac-QWACs. This design reduces complexity for Subscribers in one dimension, by removing the need to interact with and seek the multi-party approval for configuration change of a website. By reducing this complexity and dependency on third-parties, it removes a critical barrier to the adoption of QWACs present in tls-QWACs and ac-QWACs. However, it should be noted that it also introduces new challenges, in that Subscribers and Relying Parties will need to manage some of this complexity directly themselves. While these challenges are less than the proposed solutions, they may still represent an insurmountable barrier that discourages the adoption of QWACs, as a result of the cryptographic binding to individual certificates and/or keys.

Determining what data/information needs to be signed depends on identifying the security properties desired. It is difficult to make collaborative progress in the absence of a well-defined security analysis that explains the goals of QWACs, and if they are intended to be used beyond the scope of the Regulation’s stated purpose of binding identity information to a website/domain. Without such an analysis, as had previously been requested from ETSI ESI, it makes it difficult to find common understanding and ensure that the analysis accurately reflects how websites operate, and to explore whether the goals and desired properties are even achievable with web technologies. For example, in the presentation of ac-QWACs, the binding was to a single certificate. However, as demonstrated above, such a binding is insufficient and does not interoperate with web technologies, due to the plethora of certificates involved, thus it’s necessary to imagine the binding is to multiple certificates.

In this model, a cryptographic binding could be as easy as having a Subscriber sign the DER-encoded representation of the following ASN.1 pseudo-code:

```
Certificates ::= SET OF Certificate
```

```
cryptographic-QWAC-tbs ::= SEQUENCE {
  certificates [1] Certificates OPTIONAL
  ...
}
cryptographic-QWAC-data ::= SIGNED{cryptographic-QWAC-tbs}
```

In this example, the holder of the QWAC (e.g. the website operator) produces such structures, asserting what data is to be signed, similar to how CAdES provides structure for how to sign CMS-based documents.

The signature, along with the structure itself, could be provided at the previously-discussed `/.well-known/eidas` URL, along with the QWAC itself and any necessary supporting information, such as server-provided OCSP responses. A validating client could then extract this information, validate the nt-QWAC, and validate the signature over the cryptographic-QWAC-tbs. This still provides a cryptographic binding to a TLS certificate, but ensures that the website operator is able to perform the cryptographic binding themselves, without needing to involve the TSP.

The TSP would still be responsible for validating the domain names, as stated within Annex IV of the Regulation, but is otherwise wholly independent from how the Subscriber uses and manages this certificate for their domain. The cryptographic properties of the TLS certificate, or even how the certificate was validated, would be entirely orthogonal and unrelated to the QWAC. This is because a client supporting such a cryptographic binding would only validate the QWAC successfully if the domain being accessed was present within the QWAC, in addition to validating the signature over that cryptographic binding. If the QWAC was issued to a legitimate organization, they would only sign such bindings for their legitimate certificates, preventing any opportunity for misuse or confusion.

While this design demonstrates a more meaningful and alternative approach to cryptographic binding to certificates that could remove some of the identified barriers to adoption, it still presents fundamental and systemic issues with respect to privacy and how web technologies work. Thus, although this design can be said to reduce the costs, compared to those of ac-QWACs or tls-QWACs, the operational costs and complexities that remain may still represent an insurmountable barrier to widespread adoption. It is presented to emphasize that alternative designs do exist without as many of the problems of ac-QWACs or tls-QWACs.

Revisiting nt-QWACs

The approach of nt-QWACs, which avoids all of the limitations discussed within this document, thus represents the best opportunity to promote and encourage the widespread adoption of QWACs within the EU, and for which no compelling equivalent alternative has been identified.

The proposed ac-QWAC design, which rests on cryptographic binding of TLS certificates, is fundamentally limited in the technology it can support, fails to interact with the overall set of web technologies, and runs the risk of serious regressions in user privacy that may cause them to fail to work in browsers within the next several years.

The existing tls-QWAC design has already been addressed in previous replies. It faces even greater fundamental challenges to security and agility for web pages, creates greater costs to TSPs, and may evolve in directions that are fundamentally irreconcilable with browser trust stores.

The proposed nt-QWAC design manages to avoid all of these limitations, through the provision of a strong cryptographic binding to a domain or Origin, rather than a certificate. Although it is still a rough sketch of the underlying technologies, it represents a more foundational approach to providing strong assurance between a website, as represented by a domain name, and the natural or legal entity that may be operating that. It reflects the challenging lessons learned in the introduction, use, and management of OV and EV certificates, while also representing a globally-neutral, viable, interoperable path forward that minimizes the need for any changes: by TSPs, by website operators, and by software vendors.

If the goal is to ensure robust validation of website identity, as reflected by the domain name, nt-QWACs represent the most viable path identified towards promoting and encouraging their adoption. While alternatives may exist, the fundamental architectural and operational challenges identified in this document suggest that any equally-viable alternative to nt-QWACs will need to largely adopt a similar approach of eschewing a cryptographic binding to individual certificates, and focus on domains and origins.

Note that the proposal of nt-QWACs is done within the context of the Regulation and the requirements set forth by Annex IV. The requirement within Annex IV to include all of an organizations’ domain names within a QWAC, and/or to require an organization to obtain multiple QWACs for their domain names, still represents a significant challenge towards the promotion of QWACs and towards globally-interoperable solutions. This is due to the fact that, within the realm of digital services, domain names are constantly changing, reflecting different operational, product, marketing, branding, business, or security needs. As currently specified, organizations are severely limited in their ability to respond to such changes within the digital space.

In considering further updates to the eIDAS Regulation, one element worth considering is whether or not the explicit inclusion within the certificate, as set forth by Annex IV (e), is strictly necessary, or whether it may be sufficient to achieve the same effect by allowing the domain name to be expressed as part of the creation data of an electronic signature or seal. By treating the domain name as part of the creation data, much like individual documents that an organization may sign are done, it becomes possible to still achieve the constitutive effect and legal certainty, by uniquely identifying the signer, but with greater ease of use. Rather than requiring the provision of new certificates, existing electronic seal or signature certificates could be used to “self-certify”, with legal effect, their possession or operation of particular domains. While fuller technical proposals are outside the scope of this individual document, many of the lessons and challenges highlighted by this document extend beyond individual TLS connections and certificates, and apply to domains as well. The current requirements for QWACs, as set forth in Annex IV, thus represent additional barriers that otherwise don’t exist with respect to electronic seals or signatures, which may explain some of the barriers to adoption both by implementing software vendors and organizations.