FIDO Transaction Confirmation White Paper

August 2020

Editors:
Rolf Lindemann, Nok Nok Labs
Bill Leddy, VISA

Abstract

Besides generic session authentication, there is an increasing need to gather explicit user consent for a specific action, i.e. "Transaction Confirmation". Transaction Confirmation allows a relying party to not only determine if a user is involved in a transaction, but also confirm that the transaction is what the user actually intended. Transaction Confirmation achieves this by sending the details of a transaction, as seen by the relying party, to the user, and collecting a response from the user for that individual transaction. Transaction Confirmation works with any transaction, including new session instantiation (Sign-In), but provides the most value for in-session authentication.

Examples of use cases to gather user consent include:

1. Consent to pay $1000 to company X for purchasing product Y.
2. Consent to transfer $2000 from account A to account B; in particular, the European PSD2 Directive requires that the amount and payee are displayed to the payer during all phases of the corresponding online payment transaction and that the authorization becomes specific for those transaction details (called dynamic linking).
3. Consent to share specific data with a specific third party. For instance, the European GDPR Regulation mandates that a natural person provides consent to process their personal data such as:
    a. Authorization for consulting physician X to access recent blood test results from their central healthcare record,
    b. Authorization for bank X to share their contact data with insurance subsidiary Y.
    c. Authorization for identity provider X to confirm to requesting web portal Y that they are an adult.
4. Authorization of trust service provider X to remotely sign a contract with hash value H on their behalf, a typical use case occurring in online contracting under the European legal eIDAS framework.
5. Consent to sign in on a different device with a given timestamp, device name/type and IP geolocation.

In comparison to generic session authentication, the relying party has a need to:

1. Know that the user has actually seen and agreed to the action (non-repudiation), and
2. Verify that the user has given consent to that specific action at a later point in time (auditability).

Today solutions are typically either:

1. At the low end of the security spectrum: Presenting some prompt to the user via a mobile application or a web application and asking the user to click an "accept" button, or
2. At the high end of the security spectrum: Asking the user to provide a cryptographic signature or a legally binding electronic signature in jurisdictions where these are defined.

Solution (1) is easy to perform by the user, but it provides only limited security, as (i) it is difficult to know whether the user has actually seen the action they should agree to, and (ii) verifying the result at a future date depends on the long-term integrity of archived log files/audit trails, which make it difficult to demonstrate that the data has not been altered by malicious actors.

FIDO protocols specify the concept of Transaction Confirmation (with the exception of U2F) allowing a standardized and secure way of gathering explicit user consent for a specific action. This paper details the benefits of Transaction Confirmation and is a call for relying party feedback on how this can bolster related commercial initiatives – which in turn will be provided to platform vendors as they consider how best to support.

Contents

Business Needs for User Consent

There are several factors driving the need for transaction confirmation support.

Regulation

The European Banking Authority (EBA) Payment Services Directive 2 (PSD2) [1] mandates Strong Customer Authentication (SCA) for all remote transactions that are more than EUR 30. SCA requires the use of dynamic linking [2], which means multifactor authentication at the transaction, not just the session. This ensures that the authentication elements dynamically link the transaction to an amount and a payee specified by the payer when initiating the transaction. FIDO Transaction Confirmation can directly address this need. In particular, FIDO Transaction Confirmation enables securely displaying transaction details (amount and payee) during all phases of an online payment as it is mandated by the PSD2 Regulatory Technical Standards (RTS) [3].

The European General Data Protection Directive (GDPR) for protection of natural persons with regard to the processing of personal data mandates that a data subject (natural person) gives consent to processing their personal data for a specific purpose, while consent means a freely given, specific, informed and unambiguous indication of the data subject's wishes by which they, by a statement or by a clear affirmative action, signify agreement to the processing of personal data relating to them. FIDO Transaction Confirmation can directly cover this need by specifically enabling transparent rendering of the corresponding data processing purpose. As a consequence of GDPR, user consent becomes an important function concerning a wide range of online activities. It becomes even more crucial when dealing with highly sensitive personal data like the healthcare scenario cited above.

In the U.S., HIPAA does not currently require individual consent for the sharing of patient records [4], but the CARIN Alliance [5] is working with government, healthcare service providers and insurance companies to enable secure digital identities. Healthcare in the U.S. and other nations should be following the lead of the EU.

> *"Acquiring some type of electronic consent is definitely best practice and strongly preferred by the CARIN community"* – CARIN Alliance

The European Regulation on Electronic Identification and Trust Services (eIDAS) [6] specifies requirements for electronic signatures to be legally binding. Users could carry a secure signature creation device and create the signature locally, or they could use a remote service to create the signature. Remote signing of documents has become a popular use case under the European Regulation on electronic identification and trust services (eIDAS) due to its capacity for supporting mobile signing when combined with an appropriate user device. With that approach, the user authorizes a cloud-hosted hardware security module to create a legally binding signature. Legacy authentication methods (e.g. username/password + SMS-OTP) are often used to authenticate the user to the cloud service. FIDO Authentication is a more convenient and secure approach to authenticate users to the Server Signing Application [7].

With FIDO Transaction Confirmation, the Qualified Trust Service Provider (QTSP) operating the Server Signing Application has an easy and consistent way to gather user consent for the specific contents to be signed with the required non-repudiation. Without that, the contents would have to be displayed by a mobile app or a web browser and alternative measures to ensure that the user has seen the correct contents would need to be implemented. This is challenging in web browsers because of potential JavaScript injection attacks.

The Financial Data Exchange (FDX) [8] is a nonprofit dedicated to unifying the financial industry around a common, interoperable, royalty-free standard for secure and convenient consumer and business access to their financial data. FDX exists as an independent subsidiary under the umbrella of the Financial Services Information Sharing and Analysis Center (FS-ISAC), whose mission is to ensure the resilience and continuity of the global financial services infrastructure. The FDX recently published a new white paper [9] laying out the group's core principles: Five Principles of Data Sharing: Control, Access, Transparency, Traceability and Security. Among them, the following three principles could benefit greatly from the non-repudiation and auditability attributes of FIDO Transaction Confirmation:

- **Control**: Consumers should be able to effortlessly grant, modify and revoke access to their financial data for applications or services they desire to use.
- **Transparency**: Individuals using financial services should know how, when, and for what purpose their data is used and know who they have permissioned.
- **Traceability**: All data transfers should be traceable. Consumers should have a complete view of all parties that are involved in the data-sharing flow.

A variety of attacks including man-in-the-browser, remote Trojans, cross-site scripts, JavaScript injection, and session hijacking can be used to create or manipulate transactions. The consumer cannot easily detect these attacks, so they think they are safe, but they can be tricked into paying the wrong amount or paying the wrong party. For example, banking Trojans like Anubis or Panda can provide remote access to an account holder's system and allow an attacker to initiate a payment in an existing session. Transaction Confirmation should thwart these attacks by binding the authentication to the transaction through a user gesture and letting the consumer trust the amount and payee.

The FIDO UAF and PKI in Asia Case Study and Recommendations [10] white paper identified use cases across Asia where digital signing using PKI could be complemented with FIDO. These use cases include legal documents and mandatory signing for high risk transactions. FIDO Transaction Confirmation can improve security and convenience to assist adoption.

Regulatory Requirements on Auditability

Regulations typically require financial institutions to prove compliant operation in audits (e.g. for PSD2, see PSD2: Effectiveness of RTS on SCA resulted in an audit requirement [32]). Part of such audits is to prove that a specific user has authorized a specific transaction. When using transaction confirmation, this process is simplified as the assertion is a cryptographically signed object that is tamper-evident, meaning that it is easy for an auditor to verify this object was not altered. It needs to be shown that the cryptographic key that was used for transaction confirmation was (a) under the full control of the specific user and that (b) the user actually had the ability to see the transaction text. In the case of Transaction Confirmation, (a) can be shown by presenting the records of the Authenticator registration and how the user was verified. To know whether the user had the ability to see the transaction text requires knowledge about the entity that actually displayed the text and prevented potential attackers from overlaying it with malicious contents. If the authenticator attestation data from the registration response can contain the model name of the authenticator, the FIDO Metadata Service can be used to understand its security characteristics and level of support of the transaction confirmation display.

Transaction Friendly Fraud

One study "estimates that annual chargeback in the United States amounts to 5-10% of online transactions" with 50% of chargebacks attributed to friendly fraud [11]. Friendly fraud has two main flavors:

1. A family member or friend makes a purchase without the card owner's permission.
2. The cardholder made the purchase but decides they don't want to pay for their purchase when the bill comes. These are sometimes called "liar buyers".

For each dollar of chargeback, the merchant pays almost $3 [12].

It is estimated that "friendly fraud will reach about $15 billion in 2020" in the U.S. alone [11]. This cost is borne by honest customers while people gaming the system benefit.

Transaction Confirmation addresses friendly fraud in several ways:

1. Unauthorized transactions by a friend or family member will be largely eliminated. If a friend does make a transaction, the cardholder is complicit in enabling the account access. For example, an account holder could share their authenticator with a family member or allow them to attach another authenticator to the account.
2. Liar buyers cannot credibly deny that they performed the transaction when their biometric on their device confirmed the transaction.
3. Buyers cannot dispute the amount of a transaction because they explicitly agreed to the amount.

Increased Mobile Device Use

Transaction confirmation must apply to native mobile applications as much as browsers. In the next few years, biometric support on mobile devices is expected to reach 90-95% in G20 countries. It is expected that 2.5 billion people will make remote mobile retail payments by 2024 and "will result in over 60 billion biometrically-verified transactions in 2024" with a value over $1.5T [13]. The value of this market makes it a tempting target for transaction fraud.

Binding Agreements

Strong Customer Authentication (SCA) is an essential enabler for secure Digital Identities. Digital Identities can enable online agreements across multiple parties, but these agreements cannot be binding if parties can claim they did not sign. Explicit Transaction Confirmation or "digital signing" of an agreement with FIDO will provide non-repudiation. The combination of Transaction Confirmation, FIDO Authentication, Identity and public ledger technology has the potential to make paper-based agreements obsolete. Online Binding Agreements are the clear future path, but that path relies on Transaction Confirmation to be realized.

Existing Approaches and Challenges

Today we mainly see five different approaches for gathering user consent:

1. A web application establishes a session (e.g. through a Cookie or similar) and presents some consent dialog. Once the consent gesture is given, the web application sends an update to the server to remember the consent related to that session. This approach is often used for Cookies and similar "low value" consent.
2. A user authenticates to a web application and prepares a transaction. The web application sends the demand for user consent to the web application server that in turn triggers a text message (SMS) or push notification being sent to a device related to the user that has been authenticated. The user then sees and confirms the transaction text via their mobile device.
3. A native application authenticates the user, the user prepares some transaction, and the native application presents the transaction text to the user. The user can confirm the transaction through the press of a button and the application sends the transaction to the server for further processing.
4. A native application authenticates the user, the user prepares some transaction, and the native application presents the transaction text to the user via **Android Protected Confirmation** [14]. The user can confirm the transaction through the press of a button and the application sends the transaction to the server for further processing.
5. Dedicated hardware tokens with integrated display [15], [16].

In many cases, some risk scoring is performed by the server before the transaction is further processed.

The security level of these approaches depends on several factors:

1. It expects the client side application (e.g. JavaScript + HTML in the case of web applications or the native application) that sends some transaction to the server to have performed all the tasks as expected (e.g. display the appropriate text to the user, correctly represent the user's consent response, etc.). Unfortunately, it is not trivial for a server to have certainty about the application integrity from which it receives data.
2. It expects the client platform to let the user see what the application displays and also to correctly understand which gesture to provide for approving (or rejecting) the transaction. This sounds trivial, but it is difficult when overlay attacks or similar attacks need to be considered.
3. It expects that the data being sent by the application is also received by the server – protecting against tampering by potential man-in-the-middle attacks.

These factors are especially challenging for an implementation within web applications [17], as there is no strong evidence of the HTML or JavaScript code being executed by the web browser, nor is there strong evidence of support for modern protection measures like Subresource Integrity (SRI) or Content Security Policy (CSP); even if modern protection measures are supported, the relying party still does not get strong evidence about the security policy that was actually applied by the browser (the intended policy might have been loaded through and modified by a man-in-the-middle).

But there are also challenges for implementations within native applications. Only a few platforms provide a way for a relying party to remotely receive evidence about the client side app and its integrity (Android SafetyNet [18] is one example). Today those systems typically rely on the operating system kernel integrity [19]. As a result, malware with root privileges could render these protections useless.

From this perspective, Android Protected Confirmation [14] provides a very high security level as it leverages a hardware protected user interface TrustedUI on the user's existing devices. Unfortunately, this method is only available on a small number of devices and it is not available through the W3C WebAuthn specification.

Transaction Risk Scoring

Today merchants and card issuers score potential transaction fraud based on a combination of factors including device information, IP address and account behavior. This approach is effective for existing devices on existing accounts, but falls short for new devices, new accounts and unusual behavior. This results in fraud or a poor consumer experience. When potential fraud is detected, transactions may be declined or step-up authentication may be requested.

Step-up authentication is often a one-time pin (OTP) sent via SMS or email. The OTP is then entered to complete the transaction. Unfortunately, there are several attack scenarios that can easily compromise this approach, including the following:

- Social engineering
- Email account takeover
- SIM swapping
- Spoof websites
- SS7 eavesdropping on text messages [20]
- Man-in-the-browser-attack

These attacks can result in fraudulent transactions or account compromise. Because of these weaknesses, NIST has proposed phasing out SMS OTP and it will not be a suitable approach for the future of transactions [21].

Using FIDO Authentication to log in to a site can establish the user's credentials for the session. This is a good start for improved security, but it is incomplete for transactions. Attacks can still capture payment information and manipulate transactions, and merchants will not dispute friendly fraud, as discussed above. Transaction confirmation can provide better security for the consumer and non-repudiation for the merchants and payment service providers.

By enabling Transaction Confirmation, FIDO platform and browser vendors can align industry initiatives that include W3C Web Payments, EMVCo 3DS & SRC, PSD2 / GDPR, and RTP. This will provide better security and convenience for financial transactions and the next generation of identity use cases.

FIDO Transaction Confirmation

The FIDO specifications already include Transaction Confirmation [22], [23]. This allows the relying party to include a human-friendly representation of the transaction that is then cryptographically linked to the FIDO assertion by the FIDO Authenticator.

FIDO Transaction Confirmation allows various assurance levels [24] to implement the Transaction Confirmation Display in order to allow for implementations on various devices using different implementation approaches for reaching ubiquity:

1. The transaction text or image could be displayed by some privileged software, e.g. the Authenticator Specific Module (ASM). In this case, the cryptographic hash of the transaction text will be transferred to the authenticator to be cryptographically bound to the assertion. This method is only used by authenticators that do not implement a Transaction Confirmation Display [25]. The FIDO Metadata Statement [26] expresses that accordingly. Note that in the case of FIDO2/WebAuthn, the browser or the underlying operating system typically implements the ASM.
2. The transaction text could be displayed by the Authenticator itself. In this case the implementation security can be attested to the relying party (through the attestation response generated by the authenticator). The Authenticator will also compute the cryptographic hash of the transaction text and bind it to the assertion. The security characteristic of the authenticator is included in the related FIDO Metadata Statement [26]. The following differing security approaches [27] are supported here:
    a. The authenticator could use hardened software to implement such a feature on arbitrary platforms. This is practical for platform authenticators on any platform. From a security perspective, this approach is very similar to approach (1) above.
    b. The authenticator could leverage the GlobalPlatform-specified Trusted User Interface (Trusted-UI [28]) feature [29]. This is practical for platform authenticators in smartphones and smart watches.
    c. The authenticator could use dedicated hardware for displaying the transaction. This is possible for roaming authenticators that come with their own display and (soft-) keyboard.

In both cases, the authenticator will indicate user verification as normal. In typical cases, the relying party might want to ensure user verification has happened before trusting the transaction confirmation result.

In general, in the case of "security keys", option (a) is the more likely one to actually implement gathering of user consent.

Note that Transaction Confirmation does not depend on the user to understand the difference between "normal" and "secure" displays (that would have to rely on a "Trust Indicator"), since only the interaction via the "secure" one would be interpreted as a "user gesture" triggering the cryptographic signature.

FIDO Transaction Confirmation for Native Apps

In FIDO UAF [22], the authenticator can implement support for Transaction Confirmation, in order to protect against the misuse of authenticated sessions (e.g. MITB attacks). With Transaction Confirmation, the user sees the AppID and transaction text and decides whether or not to perform an action.

A user initiates a transaction with the relying party, which generates a request containing a challenge and the transaction data. Currently, FIDO UAF supports either image or text transactions [30]. The request is sent via the FIDO Client and the ASM to the authenticator. Either the ASM or the authenticator displays the transaction to the user [25]. Depending on the authenticator, different levels of security can be used to display the transaction. If the user confirms the transaction, then the authenticator signs the response, which includes a hash of the transaction content. This is sent to the relying party for verification.

Today, FIDO UAF deployments support transactions within Native Mobile Apps and Smartwatch Apps.

Value of Adding Transaction Confirmation to Browsers

Usability Aspect: Many users use their browser for triggering security relevant actions, e.g. transferring money from one bank account to another etc. Many financial institutions in the EU send push notifications to the mobile banking app in order to display and get approval for the transaction text. For the user, this approach is not ideal as it requires the user to (1) get the mobile device, (2) unlock it, (3) click on the notification, (4) re-read the transaction text and (5) finally to approve (or reject) it.

Having web browsers implementing support for FIDO Transaction Confirmation would enable companies to directly ask for the transaction approval on the main transaction device – avoiding the need to deal with a second device.

Auditability Aspect: Because of their distributed and dynamic nature, web applications with all their third party contents are more complex from an auditing perspective, see "Existing Approaches and Challenges". Transaction Confirmation-supporting web browsers would obviate the need to have the audit trail include the chain of HTML, CSS and JS code involved in the application. Instead, the signed assertion cryptographically including the transaction text and potentially an indication of the web browser are sufficient to know that the user has seen and approved the transaction text – assuming that the client system was not badly compromised.

Verifying Correct Relying Party

One issue with transactions is allowing the user to know that they are completing a transaction with the correct relying party. For example, if a shopper thinks they are on ecommerce.com but it is actually a spoof site, Transaction Confirmation can provide a hint that they are really shopping at nocommerce.com before they confirm the transaction.

Rendering the relying party transparent during transaction confirmation represents (in combination with specific key selection for the given relying party) an effective means to fight against phishing attacks. It therefore represents a strategic advantage of FIDO over other alternative approaches, in particular regarding online banking. Phishing attacks have recently increased significantly and thus become an important global threat.

To make the displayed relying party information reliable, it should be provided by the entity owning the authenticated session (e.g. the browser or the app) or by the Authenticator (in combination with the relying party key selection process) and specifically be marked when displaying, while the latter is important to enable the user to distinguish this information from possibly fake relying party information that could be sent by the server-side.

One approach that is used in FIDO UAF is to display the AppID [25] (or TLD+1). For example, "Approve Transaction from ecommerce.com" versus "Approve Transaction from nocommerce.com". The other protection measure specified by FIDO is the use of the relying party-specific "signing key". This means that the FIDO Authenticator will automatically select the FIDO credential for "ecommerce.com" for the ecommerce.com transaction and to select the FIDO credential registered for "nocommerce.com" in the case of nocommerce.com triggering the transaction. Users will not typically have registered with malicious relying parties like nocommerce.com.

In addition to key selection being specific to a given relying party and thus providing an implicit protection against fake sites, rendering the relying party transparent during transaction confirmation represents a complementary effective means to fight against phishing attacks. It consequently represents a strategic advantage of FIDO over alternative approaches, particularly with regard to online banking use cases.

For the purpose of making the displayed relying party information reliable, it should be provided by the token (in combination with the relying party key selection process) and specifically be marked as "token-provided information" for enabling the user to distinguish this evidence from possibly fake relying party information that could be sent by the server-side.

Call to Action

If you have business needs for Transaction Confirmation, you can use FIDO Transaction Confirmation in native mobile apps today.

If you want to use this concept directly through web browsers, please contact us [31] and tell us about your use cases.

References

[1] European Banking Authority Payment Services Directive 2 (PSD2), November 2015, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32015L2366

[2] Regulatory Technical Standards on Strong Customer Authentication and common and secure communication under Article 98 of Directive 2015/2366 (PSD2), November 2017, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32018R0389

[3] European Banking Authority, Final draft RTS on passporting, December 2016, https://eba.europa.eu/sites/default/documents/files/documents/passporting20(EBA-RTS-2016-08).pdf

[4] U.S. Department of Health and Human Services, What is the difference between "consent" and "authorization" under the HIPAA Privacy Rule?, July 2013, https://www.hhs.gov/hipaa/for-professionals/faq/264/what-is-the-difference-between-consent-and-authorization/index.html

[5] CARIN Alliance, https://www.carinalliance.com/

[6] European Regulation on Electronic Identification and Trust Services (eIDAS), September 2015, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32015R1502&from=EN

[7] FIDO Alliance, Using FIDO with eIDAS Services - Deploying FIDO2 for eIDAS QTSPs and eID schemes White Paper, April 2020, https://fidoalliance.org/white-paper-using-fido-with-eidas-services/

[8] Financial Data Exchange, https://financialdataexchange.org/

[9] Financial Data Exchange, The Five Principles of Data Sharing White Paper, August 2019, https://financialdataexchange.org/common/Uploaded%20files/10.3_FDX_WhitePaper_Final.pdf

[10] FIDO Alliance, FIDO UAF and PKI in Asia – Case Study and Recommendations, December 2018, https://fidoalliance.org/fido-uaf-and-pki-in-asia-case-study-and-recommendations/

[11] Mercator Advisory Group, Merchant Chargebacks Are on the Rise Due to Friendly Fraud, December 2019, https://www.mercatoradvisorygroup.com/Reports/Merchant-Chargebacks-Are-on-the-Rise-Due-to-Friendly-Fraud/

[12] ClearSale, Chargeback Fees: What Do Chargebacks Cost?, May 2019, https://blog.clear.sale/chargeback-fees-what-do-chargebacks-cost

[13] Juniper Research, Mobile Payment Authentication: Biometrics, Regulation and Forecasts 2020-2024, January 2020, https://www.juniperresearch.com/researchstore/fintech-payments/mobile-payment-authentication-market-research/subscription/biometrics-regulation-and-forecasts

[14] Android Protected Confirmation, May 2020, https://source.android.com/security/protected-confirmation

[15] Thales Group, Social engineering attacks in Corporate Banking: PKI and WYSIWYS solutions, https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/digital-banking/on-the-job-pki

[16] OneSpan, Digipass 785 Bluetooth Smart-enabled device for smartphones and tablets protects data in online, enterprise or mobile transactions, October 2018, https://www.onespan.com/sites/default/files/2019-09/digipass-785-datasheet_tcm42-73833.pdf

[17] A Framework for Web Application Integrity, January 2018, https://www.scitepress.org/Papers/2018/67202/67202.pdf

[18] Android SafetyNet Attestation API, December 2019, https://developer.android.com/training/safetynet/attestation

[19] SafetyNet: Google's tamper detection for Android, September 2015, https://koz.io/inside-safetynet/

[20] SS7 Attack, July 2016, https://whatis.techtarget.com/definition/SS7-attack

[21] NIST SP 800-63b Digital Identity Guidelines: Authentication and Lifecycle Management, July 2020, https://pages.nist.gov/800-63-3/sp800-63b.html

[22] FIDO UAF Protocol Specification, November 2017, https://fidoalliance.org/specs/fido-uaf-v1.2-rd-20171128/fido-uaf-protocol-v1.2-rd-20171128.html#transaction-confirmation

[23] W3C WebAuthn Simple Transaction Authorization Extension (txAuthSimple), March 2019, https://www.w3.org/TR/webauthn/#sctn-simple-txauth-extension

[24] FIDO Alliance, Certified Authenticator Levels, https://fidoalliance.org/certification/authenticator-certification-levels/

[25] FIDO Alliance, FIDO UAF Authenticator Commands, Sign Command, November 2017, https://fidoalliance.org/specs/fido-uaf-v1.2-rd-20171128/fido-uaf-authnr-cmds-v1.2-rd-20171128.html#sign-command

[26] FIDO Alliance, FIDO Metadata Statements, February 2018, https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-metadata-statement-v2.0-id-20180227.html

[27] "Why Scalable Attacks Matter", publication in DuD 4/2016, January 2016, https://www.springerprofessional.de/en/avoiding-the-tsunami/10162734

[28] GlobalPlatform, Trusted User Interface API v1.0, June 2013, https://globalplatform.org/specs-library/trusted-user-interface-api-v1/

[29] Rolf Lindemann, "Trusted UI and its role in protecting against malware attacks", paper presented at ARM TechCon in 2014

[30] FIDO Alliance UAF Protocol Specification Transaction Dictionary, November 2017, https://fidoalliance.org/specs/fido-uaf-v1.2-rd-20171128/fido-uaf-protocol-v1.2-rd-20171128.html#transaction-dictionary

[31] info@fidoalliance.org

[32] Deloitte, PSD2: Effectiveness of RTS on SCA resulted in an audit requirement, https://www2.deloitte.com/cz/en/pages/financial-services/articles/smernice-psd2-povinne-audity.html