United States Patent Application Publication	20250267145
Kind Code	A1
Publication Date	August 21, 2025
Inventor(s)	RULE; Jeffrey et al.

SYSTEMS AND METHODS FOR DEVICE BINDING AUTHENTICATION

Abstract

The disclosed systems and methods are directed to generating a GPU-based mobile device signature to enhance the strength of an OTP card authentication signal. The proposed implementation leverages the NFC read capability of contactless OTP cards and WebGL image rendering functionality of mobile browser. An image, or a URL pointing to one, is received, via NFC transmission from a contactless card, for processing by a mobile browser. The output of the mobile browser image processing buffer can then be hashed and used as a device identifier for the specific mobile device performing electronic authentication of a transmission source by verifying, with high degree of certainty, the identity of the reading mobile device.

Inventors: **RULE; Jeffrey (Chevy Chase, MD), OSBORN; Kevin (Newton Highlands, MA)**

Applicant: **Capital One Services, LLC** (McLean, VA)

Family ID: **1000008586747**

Appl. No.: **19/202221**

Filed: **May 08, 2025**

Related U.S. Application Data

parent US continuation 18118987 20230308 parent-grant-document US 12335256 child US 19202221

Publication Classification

Int. Cl.: **H04L9/40** (20220101)

U.S. Cl.:

CPC **H04L63/0838** (20130101);

Background/Summary

CROSS-REFERENCE TO RELATED APPLICATION [0001] This patent application is a continuation of U.S. patent application Ser. No. 18/118,987, filed Mar. 8, 2023, the contents of which are incorporated herein by reference in their entirety.

FIELD OF THE DISCLOSURE

[0002] The present disclosure is generally related to secure electronic authentication, and more specifically to a graphics processing unit based device signature generation for providing a user device authentication.

BACKGROUND

[0003] As device manufactures continue to restrict access to information specific to a device configuration and operation, it has become increasingly difficult to generate unique device identifiers that may be persistently associated with a particular device initiating a secure electronic transaction. These unique features can be used to validate a device in consecutive authentication transactions. Blocking of access to device-specific data impedes the integration of device fingerprint data, for reliable identification of a source device as a verification factor, in user authentication systems and processes thus hampering efforts for enhancing access verification security by associating a device to previous authentication transactions for a specific user.

[0004] These and other deficiencies exist Therefore, there exists a need for a device signature and/or fingerprint computation system and process that is both accessible and can be readily integrated into a user authentication process.

SUMMARY OF THE DISCLOSURE

[0005] Embodiments of the present disclosure provide a system, method, and non-transitory computer-accessible medium having stored thereon computer-executable instructions for implementing a device binding authentication system and process. In some aspects, the techniques described herein relate to a method for enhancing one time password (OTP) card authentication based on computation of a digital signature associated with operation of a graphics processing unit (GPU) of a mobile and/or computing device and providing the computed GPU-based device fingerprint as authentication data for validating a transaction-initiating source device and/or a transacting user's identity. The method includes: receiving an authentication message, via a near field communication (NFC) transmission, by a mobile device from a contactless card, the authentication message, corresponding to a user authentication request, being operative to provide raw image data associated with an image, to a mobile browser running on the mobile device; rendering, by the mobile browser, the image associated with raw image data, using a web graphics library (WebGL) application programming interface (API); generating an image hash identifier from rendered image data generated by the mobile browser using the WebGL API, the image hash identifier corresponding to the user authentication request; mapping the image hash identifier with a graphics processing unit (GPU) associated with the mobile device, for binding the mobile device to the authentication message provided by the NFC transmission from the contactless card; comparing, by an authentication server, the image hash identifier received from the mobile device, to one or more previously stored hash identifiers associated with one or more previous user authentication requests; verifying, by the authentication server, the mobile device based on determining a match between the image hash identifier received from the mobile device in response to the user authentication request, and the one or more previously stored hash identifiers associated with the one or more previous user authentication request.

[0006] In some aspects, the techniques described herein relate to a method, wherein the authentication message include a uniform resource locator (URL) pointing to the image, the image being hosted on a web server, and the mobile device, responsive to receiving the URL, being operative to retrieve raw image data from the web server. The raw image data associated with the image may be retrieved from the web server through a URL redirection. In some embodiments, the URL may include embedded instructions for redirecting to multiple images to be periodically rotated, the multiple images being stored on the web server. Alternatively, the multiple images may be stored on one or more distinct web servers.

[0007] In some aspects, the techniques described herein relate to a method, wherein raw image data is stored in an near field communication data exchange format (NDEF) file on the contactless card, the NDEF file further including an image identifier corresponding to a multipurpose internet mail extensions (MIME) media type of the image for facilitating the rendering of the image by the mobile browser having a web graphics library (WebGL) application programming interface (API). The NDEF file may be directly transmitted to the mobile device for rendering, via the NFC transmission from the contactless card. With respect to the aforementioned scenario, NFC transmission of the authentication message may be initiated by conducting an NFC read of the contactless card by a NFC reader application, running on the mobile device.

[0008] In some embodiment, the raw image data may be directly read from the contactless card by a website via web near filed communication (WebNFC) and transmitted, via a network connection, from a web server hosting the WebNFC-enabled website, to a mobile browser, on the mobile device, for rendering. The raw image data may be associated with a high entropy pattern to exaggerate GPU differences in the rendered output (e.g., the rendered image data). In accordance with some embodiments, the image may be rendered by the mobile browser in a fixed size frame buffer to prevent changes in rendered image data resulting from different screen resolutions.

[0009] In some aspects, the techniques described herein relate to a multi-factor authentication system based on integrating device binding functionality with OTP authentication card, the system including a computer hardware arrangement configure to: provide a first image data associated with an image, to a mobile browser running on a mobile device of a user, the first image data received, as part of an authentication message, in response to a user authentication request, from a contactless card associated with the user; render the image from the first image data using a web graphics library (WebGL) functionality associated with the mobile browser, to generated a second image data; generate an image hash identifier from the second image data; map the image hash identifier with a graphics processing unit (GPU) associated with the mobile device to bind the mobile device with the authentication message transmitted from the contactless card; compare, by an authentication server, the image hash identifier received from the mobile device, to one or more previously stored hash identifiers associated with one or more previous user authentication requests; verify, by the authentication server, the mobile device based on determining a match between the image hash identifier received from the mobile device in response to the user authentication request and the one or more previously stored hash identifiers associated with one or more previous user authentication request.

[0010] In some aspects, the techniques described herein relate to a system, wherein the system is further configured to encode, into the authentication message, a uniform resource locator (URL) pointing to an image hosted on a web server, the URL directing the mobile browser to retrieve the first image data from the web server. The URL may include embedded instructions for redirecting to multiple images to be periodically rotated, the multiple images being stored on the web server.

[0011] In some aspects, the techniques described herein relate to a system, wherein the first image data is stored on the contactless card and transmitted to the mobile device, for rendering, via a NFC transmission from the contactless card, the NFC transmission further including an image identifier corresponding to a multipurpose internet mail extensions (MIME) media type of the image, to facilitate the rendering of the image by the WebGL functionality of the mobile browser.

[0012] In accordance with some embodiments, the first image data may correspond to raw image data having a high entropy pattern to exaggerate GPU differences in generating the second image data, the second image data corresponding to rendered image data.

[0013] In some aspects, the techniques described herein relate to a non-transitory computer-accessible medium including instructions for execution by a computer hardware arrangement, wherein, upon execution of the instructions the computer hardware arrange is configured to perform procedures including: receiving an authentication message, via a near field communication (NFC) transmission, by a mobile device from a contactless card, the authentication message, corresponding to a user authentication request, being operative to provide raw image data associated with an image, to a mobile browser running on the mobile device; rendering, by the mobile browser, the image associated with raw image data, using a web graphics library (WebGL) application programming interface (API); generating an image hash identifier from rendered image data generated by the mobile browser using the WebGL API, the image hash identifier corresponding to the user authentication request; mapping the image hash identifier with a graphics processing unit (GPU) associated with the mobile device to provide binding between the mobile device and the authentication message provided by the NFC transmission from the contactless card; comparing, by an authentication server, the image hash identifier received from the mobile device, to one or more previously stored hash identifiers associated with one or more previous user authentication requests; verifying, by the authentication server, the mobile device based on determining a match between the image hash identifier received from the mobile device in response to the user authentication request and the one or more previously stored hash identifiers associated with one or more previous user authentication request.

[0014] In some aspects, the techniques described herein relate to a non-transitory computer-accessible medium, further including instructions for encoding, into the authentication message, a uniform resource locator (URL) pointing to an image hosted on a web server, the URL directing the mobile browser to retrieve the first image data from the web server. In some embodiments, the non-transitory computer-accessible medium may further include instructions for redirecting to multiple images to be periodically rotated.

[0015] In some aspects, the techniques described herein relate to a non-transitory computer-accessible medium, further including instructions for rendering the image directly from the NFC transmission received from the contactless card, the NFC transmission including raw image data and a multipurpose internet mail extensions (MIME) media type associated with the image, and stored on the contactless card.

Description

BRIEF SUMMARY OF THE DRAWINGS

[0016] FIG. **1** illustrates an exemplary system implementation of an authentication process with device binding, in accordance with some embodiments of the present disclosure.

[0017] FIG. **2** illustrates an overview of a mobile browser operation in generating a device fingerprint based on an image hash identifier, in accordance with some embodiments of the present disclosure.

[0018] FIG. **3** illustrates a device-binding authentication approach based on GPU-based device fingerprinting facilitated by an image URL transmitted via NFC from a contactless card, in accordance with exemplary embodiments of the present disclosure.

[0019] FIG. **4** illustrates a device-binding authentication approach based on a GPU-based device fingerprint computed from image data directly retrieved from a contactless card, in accordance with exemplary embodiments of the present disclosure.

[0020] FIG. **5** illustrates a GPU-based mobile device verification process based on direct NFC read of a contactless card by a website using WebNFC and network transmission of input image data to the mobile device, in accordance with exemplary embodiments of the present disclosure.

[0021] FIG. **6** illustrates a flowchart of an exemplary process for device-binding authentication comprising input image data acquisition for generation and verification of a GPU-based device signature, in accordance with exemplary embodiments of the present disclosure.

[0022] FIG. **7** illustrates a timing sequence diagram for a GPU-based device binding authentication, using an NFC transmitted image URL, in accordance with exemplary embodiments of the present disclosure.

[0023] FIG. **8** illustrates a timing sequence diagram for a GPU-based device binding authentication, using raw image data stored on a contactless card, in accordance with exemplary embodiments of the present disclosure.

[0024] FIG. **9** illustrates a block diagram of an exemplary system, in accordance with exemplary embodiments of the present disclosure.

DETAILED DESCRIPTION

[0025] The following description of embodiments provides non-limiting representative examples referencing numerals to particularly describe features and teachings of different aspects of the invention. The exemplary embodiments described will be recognized as capable of implementation separately, or in combination, with other embodiments from the description of the embodiments and the features and teachings of any embodiment can be interchangeably combined with the features and teachings of any other embodiment. A person of ordinary skill in the art reviewing the description of exemplary embodiments will learn and understand the different described aspects of the invention. The description of exemplary embodiments should facilitate understanding of the invention to such an extent that other implementations, not specifically covered but within the knowledge of a person of skill in the art having read the description of exemplary embodiments, will be understood to be consistent with an application of the invention.

[0026] Furthermore, the described features, advantages, and characteristics of the exemplary embodiments may be combined in any suitable manner. A person of ordinary skill in the art will recognize that the exemplary embodiments may be practiced without one or more of the specific features or advantages of an exemplary embodiment. In other instances, additional features and advantages may be recognized in certain exemplary embodiments that may not be present in all exemplary embodiments. One skilled in the art will understand that the described features, advantages, and characteristics of any exemplary embodiment can be interchangeably combined with the features, advantages, and characteristics of any other exemplary embodiment.

[0027] One aspect of the proposed system and process is directed to a device binding authentication approach that utilizes a distinct computational flow of a graphics processing unit (GPU) to derive a device signature and/or fingerprint for verifying a source device. More specifically, the invention leverages the NFC based connectivity of cryptographic OTP authentication cards to provide an input image, read by an NFC reader, for processing on a mobile device. The image would correspond to a high entropy pattern to exaggerate GPU differences in the rendered output. The output of the image processing buffer associated with a browser application (equipped with a WebGL API) running on the mobile device, can then be hashed and used as a device identifier for the specific mobile device. The proposed device binding approach can be readily integrated with the cryptographic OTP authentication process to enhance authentication strength of an OTP card authentication signal, by verifying, with high degree of certainty, the identity of the card-reading device.

[0028] The proposed solution provides a factor of authentication strength (based on using a GPU-based device signature as a device verification signal) without requiring additional authentication actions by a user. This is of inherent value in secure electronic transaction processing. Furthermore, the inventive process may be operationally integrated with contactless OTP card technology, by leveraging the NFC nature of the encrypted OTP authentication process. This will enhance the strength of the authentication process with a verifiable GPU signature associated with the source device.

[0029] FIG. **1** illustrates an exemplary system implementation **100** for an authentication process with device binding functionality based on GPU fingerprinting (e.g., providing a unique processing signature associated with a specific GPU). In some embodiments, a GPU fingerprinting process may be implemented for determining a device signature (e.g., based on an operational signature associated with a GPU of the device) for a client device **108**. The GPU fingerprinting process **102** may be implemented as part of a WebGL-supplemented mobile browser, represented by applications **118**, on the client device **108**, as illustrated in FIG. **1**.

[0030] In some embodiments, a device signature for facilitating device-binding authentication, may be implemented as a function of the image rendering process associated with a specific GPU (e.g., GPU **116**). The image rendering process associated with GPU **116** may be performed on an input image **133** retrieved from an image storing device (e.g., image hosting server **130**) As shown in FIG. **1**, the GPU fingerprint computation process **102** may be implemented as part of browser functionality, having a WebGL extension, running on the client device **108** (e.g., a mobile device associated with a user). The exemplary system implementation **100** further comprises, a network **106**, an authentication server **110**, a database **109**, and an image hosting device/server **130**. Although FIG. **1** illustrates single instances of components of system **100**, system **100** may include any number of components.

[0031] The Authentication server **110** may include one or more processors **111**, and memory **112**. Memory **112** may include one or more applications, such as applications **114**. According to the exemplary embodiment **100**, a device signature verification process **119** may be implemented as part of applications **114** stored on the Authentication server **110**. The Authentication server **110** may be in data communication with any number of components of system **100**. For example, Authentication server **110** may be configured as a central system, server or platform to control and call various data at different times to execute a plurality of workflow actions such as verification of a device signature **120**, computed by the process **102** running on the client device **108** and transmitted to the authentication server for verification. Authentication server **110** may be configured to connect to client device **108** and image hosting device **130**. Client device **108** may be in data communication with the applications **114** running the device signature verification process **119**. For example, the client device **108** may be in data communication with applications **114** and the image hosting device **130** via one or more networks **106**. The Authentication server **110** may transmit, for example from applications **114** executing thereon, one or more requests to client device **108**. The one or more requests may be associated with retrieving a device signature **120** from the client device **108**. Client device **108** may receive the one or more requests from Authentication server **110**. Without limitation, the Authentication server **110** may be a network-enabled computer. As referred to herein, a network-enabled computer may include, but is not limited to a computer device, or communications device including, e.g., a server, a network appliance, a personal computer, a workstation, a phone, a handheld PC, a personal digital assistant, a contactless card, a thin client, a fat client, an Internet browser, a kiosk, a tablet, a terminal, an ATM, or other device. The Authentication server **110** also may be a mobile device; for example, a mobile device may include an iPhone, iPod, iPad from Apple® or any other mobile device running Apple's iOS® operating system, any device running Microsoft's Windows® Mobile operating system, any device running Google's Android® operating system, and/or any other smartphone, tablet, or like wearable mobile device.

[0032] The Authentication server **110** may include processing circuitry and may contain additional components, including processors, memories, error and parity/CRC checkers, data encoders, anticollision algorithms, controllers, command decoders, security primitives and tamper-proofing hardware, as necessary to perform the functions described herein. The Authentication server **110** may further include a display and input devices. The display may be any type of device for presenting visual information such as a computer monitor, a flat panel display, and a mobile device screen, including liquid crystal displays, light-emitting diode displays, plasma panels, and cathode ray tube displays. The input devices may include any device for entering information into the authentication server that is available and supported by the authentication server, such as a touch-screen, keyboard, mouse, cursor-control device, touch-screen, microphone, digital camera, video recorder or camcorder. These devices may be used to enter information and interact with the software and other devices described herein.

[0033] The information used by the device signature verification process **119** running, for example, on the authentication server **110**, may comprise one or more user authentication data (associated with a target user account) provided, via the client device **108**, across network **106**, and/or one or more stored device signature records (computed based on image **133**) and corresponding to previous device authentication attempts initiated from the client device **108** and transmitted to the authentication server **110** across network **106**.

[0034] In some examples, network **106** may be one or more of a wireless network, a wired network or any combination of wireless network and wired network, and may be configured to connect to any one of components of system **100**. For example, the Authentication server **110** may be configured to connect to client device **108** via network **106**. In some examples, network **106** may include one or more of a fiber optics network, a passive optical network, a cable network, an Internet network, a satellite network, a wireless local area network (LAN), a Global System for Mobile Communication, a Personal Communication Service, a Personal Area Network, Wireless Application Protocol, Multimedia Messaging Service, Enhanced Messaging Service, Short Message Service, Time Division Multiplexing based systems, Code Division Multiple Access based systems, D-AMPS, Wi-Fi, Fixed Wireless Data, IEEE 802.11b, 802.15.1, 802.11n and 802.11g, Bluetooth, NFC, Radio Frequency Identification (RFID), Wi-Fi, and/or the like.

[0035] In addition, network **106** may include, without limitation, telephone lines, fiber optics, IEEE Ethernet 902.3, a wide area network, a wireless personal area network, a LAN, or a global network such as the Internet. In addition, network **106** may support an Internet network, a wireless communication network, a cellular network, or the like, or any combination thereof. Network **106** may further include one network, or any number of the exemplary types of networks mentioned above, operating as a stand-alone network or in cooperation with each other. Network **106** may utilize one or more protocols of one or more network elements to which they are communicatively coupled. Network **106** may translate to or from other protocols to one or more protocols of network devices. Although network **106** is depicted as a single network, it should be appreciated that according to one or more examples, network **106** may comprise a plurality of interconnected networks, such as, for example, the Internet, a service provider's network, a cable television network, corporate networks, such as credit card association networks, and home networks.

[0036] As shown in the exemplary system implementation **100**, illustrated in FIG. **1**, client device **108** may include one or more processors **115** coupled to a GPU **116** and memory **117**. The client device **108** can be configured as a central system, server or platform to control and call various data at different times to execute a plurality of workflow actions. The client device **108** can be configured to connect to any component of system **100** via network **106**. The client device **108** can be a dedicated server computer, such as bladed servers, or can be personal computers, laptop computers, notebook computers, palm top computers, network computers, mobile devices, wearable devices, or any processor-controlled device capable of supporting the system **100**. While FIG. **1** illustrates a single client device **108**, it is understood that other embodiments can use multiple servers or multiple computer systems as necessary or desired to support the users and can also use back-up or redundant servers to prevent network downtime in the event of a failure of a particular server.

[0037] The client device **108** can be in data communication with the image hosting device/server **130** as well as the processor **111** of the authentication server **110**. For example, client device **108** can be in data communication with processor **111** of the authentication server **110** via one or more networks **106**. The Authentication server **110** may transmit one or more requests to the client device **108**. The one or more requests can be associated with retrieving data from the client device **108**, and may be generated in response to an authentication request from a source device (e.g., client device **108**). The client device **108** can receive the one or more requests from any component of authentication server **110**. The client device **108** can be configured to transmit the requested data to the processor **111** of the authentication server **110**.

[0038] The client device **108** can include a processor **115**. The processor **115** can be, for example, one or more microprocessors. The processor **115** can include processing circuitry, which can contain additional components, including additional processors, memories, error and parity/CRC checkers, data encoders, anti-collision algorithms, controllers, command decoders, security primitives and tamper-proofing hardware, as necessary to perform the functions described herein.

[0039] The client device **108** may include one or more applications **118** comprising instructions for execution thereon. For example, the application can reside in memory **117** of client device **108** and can comprise instructions for execution on the client device **108**. The application **118** of the client device **108** can be in communication with any components of system **100**. For example, client device **108** can execute one or more applications that enable, for example, network and/or data communications with one or more components of system **100** and transmit and/or receive data. Without limitation, the client device **108** can be a network-enabled computer. As referred to herein, a network-enabled computer can include, but is not limited to a computer device, or communications device including, e.g., a server, a network appliance, a personal computer, a workstation, a phone, a handheld PC, a personal digital assistant, a contactless card, a thin client, a fat client, an Internet browser, or other device. The functionality associated with the client device **108** may also be implemented on a mobile device; for example, a mobile device can include an iPhone, iPod, iPad from Apple® or any other mobile device running Apple's iOS® operating system, any device running Microsoft's Windows® Mobile operating system, any device running Google's Android® operating system, and/or any other smartphone, tablet, or like wearable mobile device.

[0040] The client device **108** can include processing circuitry and can contain additional components, including processors, GPUs, memories, error and parity/CRC checkers, data encoders, anticollision algorithms, controllers, command decoders, security primitives and tamper-proofing hardware, as necessary to perform the functions described herein. The client device **108** can further include a display and input devices. The display can be any type of device for presenting visual information such as a computer monitor, a flat panel display, and a mobile device screen, including liquid crystal displays, light-emitting diode displays, plasma panels, and cathode ray tube displays. The input devices can include any device for entering information into the client device that is available and supported by the client device, such as a touch-screen, keyboard, mouse, cursor-control device, touch-screen, microphone, digital camera, video recorder or camcorder. These devices can be used to enter information and interact with the software and other devices described herein.

[0041] System implementation **100** can include one or more databases **109**. The one or more databases **109** can comprise a relational database, a non-relational database, or other database implementations, and any combination thereof, including a plurality of relational databases and non-relational databases. In some examples, the databases **109** can comprise a desktop database, a mobile database, or an in-memory database. Further, the one or more databases **109** can be hosted internally by any component of system **100**, such as the authentication server **110** and/or the client device **108**. The one or more databases **109** can also be hosted externally to any component of the system **100**, by a cloud-based platform, or in any storage device that is in data communication with the authentication server **110** and the client device **108**. In some examples, the databases **109** can be in data communication with any number of components of system **100**. For example, the client device **108** can be configured to retrieve the data requested by processor **111** of the authentication server **110** from the databases **109**. Client device **108** can be configured to transmit the received data from databases **109** to the processor **111** via network **106**, the received data being responsive to the transmitted one or more requests. In other examples, the processor **111** can be configured to transmit one or more requests for the requested data to the databases **109** via network **106**.

[0042] An overview of an exemplary mobile browser operations for generating a device fingerprint based on a hashed image identifier is illustrated in FIG. **2**. The hashed image identifier may be computed by processing rendered image data, associated with an input image, with a cryptographic hash function. The rendering process of the mobile browser may utilize the graphics processing unit (GPU) of a mobile device via an integrated web graphics library (WebGL) functionality. Web graphics library (WebGL) is an application programing interface (API) used for graphics rendering that can be completely controlled by the web browser. WebGL specification allows internet browsers access to graphics processing unit (GPU) on the device which enables the GPU to be incorporated into the graphics computation performed by a web browser running on a mobile device. This facilitates GPU hardware accelerated architecture for graphics processing directly by the browser application. The output of the WebGL supplemented image rendering process will correspond to a rendered image data (e.g., a digital image) that may be stored in a frame buffer (e.g., a portion of the read access memory which contains a complete frame data intended for output to a display). In GPU accelerated computing, the raw image data may be loaded into the GPU. Whenever the rendering process encounter a compute-intensive portion of the code, then that portion of the code may be loaded and run on the GPU.

[0043] Differences in operating system type and version as well as other software and hardware difference and performance characteristics may result in different computation paths and different set of operations performed by a GPU in rendering an image. This may result in differences in the pixel output associated with the rendition of the image in compressed form. The difference in a pixel output of a GPU may be specially exaggerated if an input image being rendered corresponds to a highly entropic data pattern that is very hard to compress.

[0044] In some embodiments information regarding the WebGL version as well as information regarding the operating system version may be extracted based on the specific pixel output of a image rendering process and encoded into a hash identifier computed on the rendered image data. The hash identifier may then serve as a device fingerprint. FIG. **2** illustrates an overview of an exemplary process for enhancing one time password (OTP) card authentication with GPU-based device binding. The exemplary process **200** utilizes the NFC data transmission **202**, transmitted from a contactless card **204** to a computing device (e.g., mobile device **206**), to facilitate the image rendering process by a mobile internet browser **208** running on the mobile device **206**. FIG. **2** further illustrates the operations of the mobile browser **208** for generating a device fingerprint from rendered image data processed with a cryptographic hash function. The image rendering process performed by the mobile browser **208** may incorporate web graphics library (WebGL) functionality in order to utilize a graphic processing unit (GPU) of the mobile device (**206**) in the image rendering process, thus inserting into the rendered image data a processing signature of the corresponding GPU.

[0045] The operations of the mobile browser directed at GPU-supplemented rendering of an input image using WebGL API is illustrated in diagram **210**. Based on the process illustrated in diagram **210** an image processing signature may be derived and used as a device fingerprint for the computing device (e.g., mobile device **206**). As described, the process may be invoked in response to a near field communication (NFC) transmission **202** from the contactless card **204**. The NFC transmission **202** may include an authentication message comprising a uniform resource locator (URL) pointing to the image (e.g. raw image data) to be retrieved by the mobile browser **208**. In some embodiments the authentication message, transmitted via the NFC transmission **202**, may comprise the raw image data stored locally on an NFC tag of the contactless card.

[0046] The authentication message transmitted from the contactless card **204** may be received by a

NFC reader **210** and passed over to the mobile device **206** for processing. The NFC reader **210** may be integrated into the mobile device **206**. In some embodiments, corresponding, for example, to scenarios when a NFC reading application and/or capability is not available on computing and/or mobile device, WebNFC functionality may be encoded in a website to enable direct reading of the contactless card **204** by the website, launched, from example, on a personal computer (PC) terminal. Various embodiments for passing the raw image data **212** to a computing device by utilizing the NFC read capability of contactless (OTP) card **204** are further discussed in relation to FIGS. **3**, **4** and **5**.

[0047] Referring back to FIG. **2**, once the browser application (**208**) receives the raw image data **212**, whether directly from NFC transmission **202**, or via the URL encoded therein, the browser may utilize the computing resources provided by the graphics processing unit (GPU) to process and render the image. The GPU functionality may be accessed through a WebGL API **214** incorporated into the browser application **208**.

[0048] The output of the image rendering process (e.g., rendered image data **218**) may be written into a frame buffer **215**. The frame buffer **215** may store the rendered image data **218** that can be displayed as a digital image. The content **216** of the fame buffer **215** may then be hashed with a cryptographic hash function **220** to generate an image hash identifier **222**. As described earlier, the raw image data **212** may be associated with a high entropy data pattern to exaggerate GPU differences in the rendered output (**218**). In some embodiments the framebuffer **215** may correspond to a fixed sized buffer in order to prevent changes in rendered image data **218** resulting from different screen resolutions associated with the computing/mobile device **206**.

[0049] The image rendering process **210**, by utilizing a distinct computational flow of the GPU, may generate an output (e.g., rendered image data **218**) that possess a unique device-specific signature. Accordingly, hashing the rendered image data provides a hash identifier **222** that may serve as a GPU-based device fingerprint for verifying a source device. The hash identifier **222** (interchangeably referred to as image hash identifier) may then be mapped to the GPU of the mobile device **206** and used as a GPU-based device fingerprint for the mobile device **206**. The GPU-based device fingerprint **222** may then be transmitted, for verification, to an authentication server (e.g., back-end authentication server **224**) storing one or more user-device fingerprint records **226** associated with previous authentication requests initiated by the contactless card **204** via the mobile device **206**. In some embodiments the generated GPU-based device fingerprint may be integrated into operations of the contactless OTP authentication card (e.g., contactless card **204**) to add a factor of strength to the OTP card authentication signal.

[0050] In some examples, exemplary procedures in accordance with the present disclosure described herein can be performed by a computer hardware arrangement. Such a computer hardware arrangement can be, for example entirely or a part of, or include, but not limited to, a computer and/or processor that can include, for example one or more microprocessors, and use instructions stored on a non-transitory computer-accessible medium (e.g., RAM, ROM, hard drive, or other storage device). For example, a computer-accessible medium can be part of the memory of the systems and devices described herein and/or other computer hardware arrangements.

[0051] In some examples, a computer-accessible medium (e.g., as described herein, a storage device such as a hard disk, floppy disk, memory stick, CD-ROM, RAM, ROM, etc., or a combination thereof) can be provided (e.g., in communication with the computer hardware arrangement). The computer-accessible medium can contain executable instructions thereon. In addition or alternatively, a storage arrangement can be provided separately from the computer-accessible medium, which can provide the instructions to the computer hardware arrangement. The instructions can configure the computer hardware arrangement to execute certain exemplary procedures, processes, and methods, as described herein above, for example.

[0052] FIG. **3** illustrates an exemplary implementation of GPU-based (mobile) device authentication, initiated by a NFC transmission of an image URL **301** from a contactless card **302** to facilitate computation of a GPU-based device signature for the mobile device **304**. In some embodiment the computed device signature/fingerprint may be used as an authentications factor in a multi-factor authentication process (e.g. as indicated by multifactor authenticated connection **330** in FIG. **3**) for validating electronic data access requests and/or a merchant transaction initiated by using the contactless card **302** and the mobile device **304**.

[0053] Referring back to FIG. **3**, the computation of the GPU-based device signature may be initiated by conducting an NFC read of an authentication record, stored on the contactless card **302**, by NFC reader **306** (e.g., with a corresponding reader application, not shown in FIG. **3**, running on the mobile device **304**). The NFC-transmitted authentication records may comprise an image URL **301** which points to an image data file (e.g. raw image data) to be retrieved by the mobile device **304**. Once retrieved by the mobile device **304**, the URL may be passed to a browser application **308** running on the mobile device.) In some embodiments, an initial URL request message, by the mobile browser **308**, may be re-directed to a destination server hosting the image (e.g., storing the raw image data). This is shown by the initial URL request/response communication **310** between the mobile device **304** and a destination identified by the URL (e.g. server **312**). The image **318** may then be retrieved from the hosting server (e.g., authentication server **314**). Authentication server **314** may further store one or more data records **316** corresponding to previous hash identifiers of the image **318** associated with previous successful authentication attempts (using the device fingerprint) initiated from the mobile device **304**.

[0054] With reference to FIG. **3**, the raw image data **319** (associated with image **318**), may be retrieved from the authentication server **314**, and processed by the mobile browser **308**, running on the mobile device **304**, to generate rendered image data in a framebuffer **320**. The framebuffer data may then be hashed by a cryptographic hash process **322** to generate an image hash identifier **324** that may serve as a GPU-based device fingerprint for the mobile device **304**. The image hash identifier **324** may be transmitted to the authentication server **314** for comparing with previously stored image hash identifiers **316** associated with previous device authentication attempts using the mobile device **304**. If the comparison with a previously stored image hash identifier produces a match, signifying that the same device was used in previous authentication attempts, a device verification response **326** may be generated and transmitted to an authentication requesting server **328**. The device verification response **326** may correspond to a standalone authentication response, or it may be incorporated as part of a multi-factor authentication (e.g., multifactor authenticated connection **330**) along with other encrypted user identification data that may be stored on the contactless card **302** and transmitted along with the device signature **324**. Accordingly, the integration of a GPU-based device fingerprint (e.g., **324**) into the cryptographic authentication process associated with an OTP authentication card (e.g., **302**) facilitates a multifactor authenticated connection **330** between the mobile device (as initiated by the contactless card **302**), and a destination server **328**.

[0055] In some embodiments, the contactless card **302** may correspond to a uniquely configured OTP contactless card with an integrated processor **331** and a NFC tag **332** storing NFC transmittable user authentication data (readable, for example, by a mobile device with a reader component and running a corresponding application) The contactless card **302**) may further comprise a counter **333**, also referred to as application transaction counter (ATC), for keeping track of OTP transactions initiated by the contactless card, as well as one or more applets **334** for facilitating the generation of the OTP authentication cryptogram. In some embodiments the transaction counter value may be updated for each OTP transaction initiated by the contactless card.

[0056] In some embodiments, the URL may comprise embedded instructions for rendering to multiple images to be periodically rotated. The multiple images may be stored on a designated web server and/or multiple distinct servers.

[0057] FIG. **4** illustrates an exemplary embodiment **400** wherein the raw image data is directly stored onto the contactless card **402**, for example, as a near field communication data exchange format (NDEF) file **403**. In the embodiment **400** the NFC transmission **406** may correspond to the raw image data and an image identifier, corresponding to a multipurpose internet mail extensions (MIME) media type, stored on the contactless card **402** as an NDEF file **403**. The NDEF file comprising the raw image data and the image MIME type, is transmitted to the reader **405** of the mobile device in response to bringing the contactless card within NFC range of the mobile device with an operational NFC reader (e.g., tapping the contactless card on the reader of the mobile device.) Upon receiving the NFC transmission **406**, the raw image data **407** may be passed to and processed by the mobile browser running on the mobile device **404** to generate a rendered image data in a framebuffer **410**. The framebuffer data **412** is then hashed by a cryptographic hash process **414** to generate an image hash identifier **416** that may server as a GPU-based device fingerprint. The hashed image identifier is transmitted to the authentication server **418** to be compared with stored records **420** corresponding to previously stored hash identifiers of the NDEF image file **403** associated with previous device authentication attempts initiated from the mobile device **404**. If a match is determined, signifying that the same device was used in previous authentication attempts, a device verification response **422** is generated and transmitted, for example, to an authentication requesting server **424**. The device verification response **422** may be provide as a standalone device-binding authentication signal pertaining to mobile device **404** and/or as part of a multi-factor authentication **426** along with other encrypted user identification data that may be stored on the contactless card **402** and transmitted along with the mobile device fingerprint **416**. This can then facilitate the multifactor authenticated connection **428** to the destination server **424** based on an authenticated pairing of the contactless card **402** and the mobile device **404**.

[0058] FIG. **5** illustrates an embodiment for supporting a scenario wherein a mobile device (e.g., mobile device **502**) may not have NDEF read capability and/or an NFC application for establishing an NFC link with a contactless card **504**. In such a scenario, the input image **506** (e.g., raw image data along with an image MIME type) may be retrieved from the contactless card **504** via a direct NFC read **507** of the contactless card by a verification website **508**, using a web near filed communication (WebNFC) API **509**. WebNFC is a low-level API that provides websites the ability to read and write to nearby NFC devices. The (WebNFC-enabled) verification website **508** may be provided by the authentication server **510** and accessed, for example, through a browser application running on a personal computer **512**. The input image data **506** may then transmitted to a user mobile device (e.g., via network transmission **516**) by the authentication server **510** across a network connection **517** established between the authentication server **510** and the user mobile device. Subsequently the image data retrieved (directly from the contactless card) by the verification website (e.g., via a WebNFC process), may be sent to a registered mobile device (e.g., mobile device **514**, associated with the user), for rendering. The rendered image data **518** may then be read directly from the browser and hashed (e.g., by a cryptographic hash function **519**) to generate an image hash identifier **520**. The image hash identifier **520**, representing a digital fingerprint associated with the mobile device **514**, is transmitted to the authentication server **510** for verification against previous authentication records **522**. If a match is determined, the authentication sever **510** may determine that the received message has not been spoofed by a different device (e.g., different than mobile device **514**) that may be used by a hacker to facilitate a fraudulent user verification process and subsequently transmit a device verification response **524** to an authentication requesting entity (e.g., merchant server **526**).

[0059] In some embodiment the WebNFC-enable website **508** may be launched directly on the mobile device **514** to facilitate NFC-based retrieval of image data, via the direct NFC read **507**, from the contactless card **504**.

[0060] As descried above, WebNFC functionality may be encoded in a website to enable direct reading of the contactless card **504** by the WebNFC-enabled website, launched on a computing and/or a mobile device associated with a user. In some embodiment, WebNFC functionality may be encoded in a merchant website to enable direct reading the contactless card **504** by the WebNFC-enabled merchant website, launched on a computing and/or a mobile device associated with a user. The input image data may then be directly read from the contactless card by the merchant web server (via the WebNFC-enabled merchant website) and transmitted to a mobile device associated with a phone number that may be provided by the user initiating the transaction. The image may then be rendered by a mobile browser (using WebGL API) running on the mobile device, and a hash identifier of the rendered image, transmitted back to the merchant webserver. The merchant webserver may then transmit the hash identifier (e.g., device fingerprint) to an authentication server for verification against previous authentications records. If a match is determined, the authentication sever may send a device verification response to the merchant webserver.

[0061] FIG. **6** illustrate an exemplary operational flowchart **600** for generation and verification of a GPU-based mobile device fingerprint. The exemplary process **600** may be initiated by acquisition of an input image (e.g., raw image data to be rendered by a WebGL-enabled browser running on a mobile device.) The acquisition of the raw image data may be facilitated by a NFC transmission from a contactless card as shown in step **602**. The acquisition of the raw image data by the mobile device may be implemented by any of the operations described in steps **602**.**1**, **602**.**2** or **602**. **3**. For Example, the NFC transmission, from the contactless card to the mobile device reader, may comprise a URL pointing to a server which hosts the raw image data (e.g., step **602**.**1**) Alternatively, the NFC transmission, from the contactless card to the mobile device reader, may comprise the actual raw image data, stored along with an image identifier, in an NDEF file on the contactless card (e.g., step **602**.**2**) The NFC transmission may also be initiated between a website and the contactless card, using WebNFC and transmitted, via a network connection, to the mobile device (e.g., step **602**.**3**).

[0062] Upon retrieval of the raw image data at step **602**, the retrieved image data is rendered, using a WebGL process, by a corresponding mobile browser, at step **604**. The output of the image rendering process, associated with a specific GPU signature, is then hashed at step **606** to generate an image hash identifier representing a GPU-based device fingerprint. The image hash identifier (e.g., mobile device fingerprint) may then be transmitted to an authentication server for verification at step **608**. The Authentication server may be storing records corresponding to previous hash identifiers of the input image associated with previous authentication attempts. The verification process, at step **610**, may involve matching the received image hash identifier with one or more previously stored image hash identifiers (e.g., previous authentication records.) In some embodiment, the comparison may involve the most recently stored device fingerprint record. If a positive match is determined at step **610**, the mobile device is authenticated as a valid user device, at step **612**, and a device-factor authentication may be added to an OTP authentication process associated with the contactless card. If a match is not determined at step **610** a device mismatch notification may be generated at step **614** and transmitted back to a authentication requesting party and/or the transmitting mobile device.

[0063] FIG. **7** illustrate a timing sequence associated with an exemplary GPU-based device verification process **700**. The exemplary process **700** corresponds to a URL-directed acquisition of an input image (e.g., raw image data from a contactless card **702**), by a mobile device **704** communicatively coupled to a redirection server **706** and/or authentication server **708**. The computed GPU-based device fingerprint may then be used as an authentications factor in validating electronic data access requests and/or a user transaction with a secure system (e.g., merchant server **710**), initiated from the mobile device **704**. The exemplary process **700** may be triggered by a NFC-based reading of an image URL (e.g., a URL pointing to a raw image data file) by a reader component (with a corresponding reader application of the mobile device **704**.) The read operation may be initiated, for example, by tapping the contactless card **702** on the reader of the mobile device **704**. In accordance to some embodiments, the initial URL request may be re-directed (e.g. as indicated by request/response communication **714**) to a destination server, such as the authentication server **708**, that may be hosting the image data file. This is illustrated by the communication **716** between the mobile device **704** and the re-directed destination (e.g. authentication server **708**) for the acquisition of the raw image data to be processed on the user mobile device **704**.

[0064] The raw image data retrieved from the hosting server (e.g., authentication server **708** may then be rendered by a client browser application running on the mobile device to generate rendered image data as indicated by operation **717**. The content of a framebuffer associated with the output of the image rendering process (e.g., mobile browser's WebGL process) may then be hashed by a cryptographic hash process (e.g., operation **718**) to generate an image hash identifier that may server as a GPU-based device signature and/or fingerprint. The image hash identifier is transmitted (as indicated by the transmission **720**) to the authentication server **708** for comparison with previously stored image hash identifiers associated with previous device authentication attempts using the mobile device **704**.

[0065] Upon determining a successful match with one or more previous hash identifier during the verification process **722**, the mobile device is verified by the authentication server **708**. Following a successful verification of the mobile device **704** by the authentication server **708**, a device authentication message **724** may be transmitted to an authentication requesting entity (e.g., merchant server **710**) as a standalone device authentication response and/or as part of a multi-factor authentication supplemented with other encrypted user identification data, which may be stored on the contactless card **702** and transmitted to the authentication server for verification along with the device signature (e.g., image hash identifier **720**).

[0066] FIG. **8** illustrates a timing sequence associated with an exemplary GPU-based device verification process **800**. The exemplary process **800** corresponds to a direct acquisition of an input image (e.g., raw image data from a contactless card **802**), by a mobile device **804**, via a NFC proximity link **805** established between the contactless card **802** and a user mobile device **804**. The mobile device **804** may also be communicatively coupled to an authentication server **806**. The raw image data retrieved directly from the contactless card **802** may then be rendered by a client browser application running on the mobile device **804** to generate a rendered image data as indicated by operation **810**. The content of a framebuffer associated with the output of the image rendering process (e.g., mobile browser's WebGL process) may then be hashed by a cryptographic hash process (e.g., operation **812**) to generate an image hash identifier that may server as a GPU-based device signature and/or fingerprint. The image hash identifier is transmitted (indicated by the transmission **813**) to the authentication server **806** for comparison with previously stored image hash identifiers associated with previous device authentication attempts using the mobile device **804**.

[0067] Upon determining a successful match with one or more previous hash identifier during the verification process **814**, the mobile device is verified by the authentication server **806**. Following a successful verification of the mobile device **804** by the authentication server **806**, a device authentication message **816** may be transmitted to an authentication requesting entity (e.g., merchant server **808**) as a standalone device authentication response and/or as part of a multi-factor authentication supplemented with other encrypted user identification data, which may be stored on the contactless card **802** and transmitted to the authentication server for verification along with the device signature (e.g., image hash identifier **813**)

[0068] FIG. **9** shows a block diagram of an exemplary embodiment of a system according to the present disclosure. For example, exemplary procedures in accordance with the present disclosure described herein can be performed by a computer hardware arrangement **905**. Such computer hardware arrangement **905** can be, for example entirely or a part of, or include, but not limited to, a computer and/or processor **910** that can include, for example one or more microprocessors, and use instructions stored on a computer-accessible medium (e.g., RAM, ROM, hard drive, or other storage device)

[0069] As shown in FIG. **9**, for example a computer-accessible medium **915** (e.g., as described herein above), may comprise a storage device such as a hard disk, floppy disk, memory stick, CD-ROM, RAM, ROM, etc., or a collection thereof can be provided (e.g., in communication with the computer hardware arrangement **905**.) The computer-accessible medium **915** can contain executable instructions **920** thereon. In addition or alternatively, a storage arrangement **925** can be provided separately from the computer-accessible medium **915**, which can provide the instructions to the processing arrangement **905**. The instructions can configure the computer hardware arrangement to execute the exemplary procedures, processes, and methods, as described herein above, for example.

[0070] Further, the exemplary computer hardware arrangement **905** can be provided with or include an input/output ports **935**, which can include, for example a wired network, a wireless network, the internet, an intranet, a data collection probe, a sensor, etc. As shown in FIG. **9**, the exemplary computer hardware arrangement **905** can be in communication with an exemplary display arrangement **930**, which, according to certain exemplary embodiments of the present disclosure, can be a touch-screen configured for inputting information to the processing arrangement in addition to outputting information from the processing arrangement, for example. Further, the exemplary display arrangement **930** and/or a storage arrangement **925** can be used to display and/or store data in a user-accessible format and/or user-readable format.

[0071] As used herein, the term "card" is not limited to a particular type of card. Rather, it is understood that the term "card" can refer to a contact-based card, a contactless card, or any other card, unless otherwise indicated. It is further understood that the present disclosure is not limited to cards having a certain purpose (e.g., payment cards, gift cards, identification cards, membership cards, transportation cards, access cards), to cards associated with a particular type of account (e.g., a credit account, a debit account, a membership account), or to cards issued by a particular entity (e.g., a commercial entity, a financial institution, a government entity, a social club.) Instead, it is understood that the present disclosure includes cards having any purpose, account association, or issuing entity.

[0072] Systems and methods described herein can provide secure, retrieval of sensitive user information or enabling streamlined communication and processing of sensitive user information for example, for facilitating secure electronic transactions. Once a valid authorization response from an authenticated device and/or user has been established, the automated data retrieval and transfer system and process can permit, without limitation, financial transactions (e.g., credit card and debit card transactions), account management transactions (e.g., card refresh, card replacement, and new card addition transactions), membership transactions (e.g., joining and departing transactions), point of access transactions (e.g., building access and secure storage access transactions), transportation transactions (e.g., ticketing and boarding transactions, and other transactions.)

[0073] It is further noted that the systems and methods described herein may be tangibly embodied in one or more physical media, such as, but not limited to, a compact disc (CD), a digital versatile disc (DVD), a floppy disk, a hard drive, read only memory (ROM), random access memory (RAM), as well as other physical media capable of data storage. For example, data storage may include random access memory (RAM) and read only memory (ROM), which may be configured to access and store data and information and computer program instructions. Data storage may also include storage media or other suitable type of memory (e.g., such as, for example, RAM, ROM, programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), magnetic disks, optical disks, floppy disks, hard disks, removable cartridges, flash drives), and any type of tangible and non-transitory storage medium, where the files that comprise an operating system, application programs including, for example, web browser application, email application and/or other applications, and data files may be stored. The data storage of the network-enabled computer systems may include electronic information, files, and documents stored in various ways, including, for example, a flat file, indexed file, hierarchical database, relational database, such as a database created and maintained with software from, for example, Oracle® Corporation, Microsoft® Excel file, Microsoft® Access file, a solid state storage device, which may include a flash array, a hybrid array, or a server-side product, enterprise storage, which may include online or cloud storage, or any other storage mechanism. Moreover, the figures illustrate various components (e.g., servers, computers, processors, etc. separately. The functions described as being performed at various components may be performed at other components, and the various components may be combined or separated. Other modifications also may be made.

[0074] Computer readable program instructions described herein can be downloaded to respective computing and/or processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing and/or processing device receives computer readable program instructions from the network and forwards the computer readable program instructions for storage in a computer readable storage medium within the respective computing and/or processing device.

[0075] Computer readable program instructions for carrying out operations of the present invention may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like, and conventional procedural programming languages, such as the “C” programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider. In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, to perform aspects of the present invention.

[0076] These computer readable program instructions may be provided to a processor of a general-purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions specified herein. These computer-readable program instructions may also be stored in a computer-readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a manner, such that the computer readable storage medium having instructions stored therein comprises an article of manufacture including instructions which implement aspects of the functions specified herein.

[0077] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute on the computer, other programmable apparatus, or other device implement the functions specified herein.

[0078] The present disclosure is not to be limited in terms of the particular embodiments described in this application, which are intended as illustrations of various aspects. Many modifications and variations can be made without departing from its spirit and scope, as may be apparent. Functionally equivalent methods and apparatuses within the scope of the disclosure, in addition to those enumerated herein, may be apparent from the foregoing representative descriptions. Such modifications and variations are intended to fall within the scope of the appended representative claims. The present disclosure is to be limited only by the terms of the appended representative claims, along with the full scope of equivalents to which such representative claims are entitled. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to be limiting.

[0079] The foregoing description, along with its associated embodiments, has been presented for purposes of illustration only. It is not exhaustive and does not limit the invention to the precise form disclosed. Those skilled in the art may appreciate from the foregoing description that modifications and variations are possible in light of the above teachings or may be acquired from practicing the disclosed embodiments. For example, the steps described need not be performed in the same sequence discussed or with the same degree of separation. Likewise various steps may be omitted, repeated, or combined, as necessary, to achieve the same or similar objectives. Accordingly, the invention is not limited to the above-described embodiments, but instead is defined by the appended claims in light of their full scope of equivalents.

[0080] In the preceding specification, various preferred embodiments have been described with references to the accompanying drawings. It may, however, be evident that various modifications and changes may be made thereto, and additional embodiments may be implemented, without departing from the broader scope of the invention as set forth in the claims that follow. The specification and drawings are accordingly to be regarded as an illustrative rather than restrictive sense.

Claims

1. A method for enhancing one time password (OTP card authentication with GPU-based device binding, the method comprising: receiving an authentication message, via a near field communication (NFC) transmission, by a mobile device from a contactless card, the authentication message, corresponding to a user authentication request, being operative to provide raw image data associated with an image, to a mobile browser running on the mobile device; rendering, by the mobile browser, the image associated with raw image data, using a web graphics library (WebGL) application programming interface (API); generating an image hash identifier, from rendered image data generated by the mobile browser, using the WebGL API, the image hash identifier corresponding to the user authentication request; mapping the image hash identifier with a graphics processing unit (GPU) associated with the mobile device, for binding the mobile device to the authentication message provided by the NFC transmission from the contactless card; comparing, by an authentication server, the image hash identifier received from the mobile device, to one or more previously stored hash identifiers associated with one or more previous user authentication requests; verifying, by the authentication server, the mobile device based on determining a match between the image hash identifier received from the mobile device in response to the user authentication request, and the one or more previously stored hash identifiers associated with the one or more previous user authentication request.

2. The method of claim 1, wherein the authentication message comprise a uniform resource locator (URL) pointing to the image, the image being hosted on a web server, and the mobile device, responsive to receiving the URL, being operative to retrieve raw image data from the web server.

3. The method of claim 2, wherein raw image data associated with the image is retrieved from the web server through a URL redirection.

4. The method of claim 3, wherein the URL comprises embedded instructions for redirecting to multiple images to be periodically rotated, the multiple images being stored on the web server.

5. The method of claim 4, wherein the multiple images are stored on one or more distinct web servers.

6. The method of claim 1, wherein raw image data is stored in an near field communication data exchange format (NDEF) file on the contactless card, the NDEF file further comprising an image identifier corresponding to a multipurpose internet mail extensions (MIME) media type of the image for facilitating the rendering of the image by the mobile browser WebGL API.

7. The method of claim 6, wherein the NDEF file is transmitted to the mobile device for rendering, via the NFC transmission from the contactless card.

8. The method of claim 6, wherein raw image data associated with the image is directly read from the contactless card by a web server via web near filed communication (WebNFC) and transmitted to the mobile browser on the mobile device for rendering.

9. The method of claim 1, wherein a transmission of the authentication message is initiated by conducting an NFC read of the contactless card by a NFC reader application, running on the mobile device.

10. The method of claim 1, wherein, raw image data comprises a high entropy pattern to exaggerate GPU differences in generating rendered image data.

11. The method of claim 1, wherein the image is rendered by the mobile browser in a fixed size frame buffer to prevent changes in rendered image data resulting from different screen resolutions.

12. A multi-factor authentication system based on integrating device binding functionality with OTP authentication card, the system comprising a computer hardware arrangement configure to: provide a first image data associated with an image, to a mobile browser running on a mobile device of a user, the first image data received, as part of an authentication message, in response to a user authentication request, from a contactless card associated with the user; render the image from the first image data using a web graphics library (WebGL) functionality associated with the mobile browser, to generated a second image data; generate an image hash identifier from the second image data; map the image hash identifier with a graphics processing unit (GPU) associated with the mobile device to bind the mobile device with the authentication message transmitted from the contactless card; compare, by an authentication server, the image hash identifier received from the mobile device, to one or more previously stored hash identifiers associated with one or more previous user authentication requests; verify, by the authentication server, the mobile device based on determining a match between the image hash identifier received from the mobile device in response to the user authentication request and the one or more previously stored hash identifiers associated with one or more previous user authentication request.

13. The system of claim 12, wherein the system is further configured to encode, into the authentication message, a uniform resource locator (URL pointing to an image hosted on a web server, the URL directing the mobile browser to retrieve the first image data from the web server.

14. The system of claim 13, wherein the URL comprises embedded instructions for redirecting to multiple images to be periodically rotated, the multiple images being stored on the web server.

15. The system of claim 12, wherein the first image data is stored on the contactless card and transmitted to the mobile device, for rendering, via a NFC transmission from the contactless card, the NFC transmission further comprising an image identifier corresponding to a multipurpose internet mail extensions (MIME) media type of the image, to facilitate the rendering of the image by the WebGL functionality of the mobile browser.

16. The system of claim 12, wherein, the first image data corresponds to raw image data having a high entropy pattern to exaggerate GPU differences in generating the second image data, the second image data corresponding to rendered image data.

17. A non-transitory computer-accessible medium comprising instructions for execution by a computer hardware arrangement, wherein, upon execution of the instructions the computer hardware arrange is configured to perform procedures comprising: receiving an authentication message, via a near field communication (NFC) transmission, by a mobile device from a contactless card, the authentication message, corresponding to a user authentication request, being operative to provide a raw image data associated with an image, to a mobile browser running on the mobile device; rendering, by the mobile browser, the image associated with the raw image data, using a web graphics library (WebGL application programming interface (API); generating an image hash identifier from a rendered image data generated by the mobile browser using the WebGL API, the image hash identifier corresponding to the user authentication request; mapping the image hash identifier with a graphics processing unit (GPU) associated with the mobile device to provide binding between the mobile device and the authentication message provided by the NFC transmission from the contactless card; comparing, by an authentication server, the image hash identifier received from the mobile device, to one or more previously stored hash identifiers associated with one or more previous user authentication requests; verifying, by the authentication server, the mobile device based on determining a match between the image hash identifier received from the mobile device in response to the user authentication request and the one or more previously stored hash identifiers associated with one or more previous user authentication request.

18. The non-transitory computer-accessible medium of claim 17, further comprising instructions for encoding, into the authentication message, a uniform resource locator (URL) pointing to an image hosted on a web server, the URL directing the mobile browser to retrieve the raw image data from the web server.

19. The non-transitory computer-accessible medium of claim 17 further comprising instructions for redirecting to multiple images to be periodically rotated.

20. The non-transitory computer-accessible medium of claim 17, further comprising instructions for rendering the image directly from the NFC transmission received from the contactless card, the NFC transmission comprising raw image data and a multipurpose internet mail extensions (MIME media type associated with the image, and stored on the contactless card.