associated with the account data in response to providing the secure payment container request.

[0008] In some non-limiting embodiments or aspects, determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core.

[0009] In some non-limiting embodiments or aspects, the secure payment container request is provided in response to a successful handshake between the SDK wrapper and the SDK core.

[0010] In some non-limiting embodiments or aspects, executing the SDK wrapper exposes the real function as a public method, and the real function throws an error in response to being called by the merchant system.

[0011] In some non-limiting embodiments or aspects, the SDK wrapper includes an Accept.js library, and the SDK core includes an AcceptCore.js library.

[0012] According to some non-limiting embodiments or aspects, provided is a computing system including: one or more processors programmed and/or configured to: load, at a merchant system, a software development kit (SDK) wrapper from a payment gateway system via a merchant webpage associated with the merchant system; execute the SDK wrapper, wherein the SDK wrapper loads an SDK core when executed; determine an integrity of the SDK core; in response to determining the integrity of the SDK core, perform a handshake between the SDK wrapper and the SDK core and overloading a real function exported by the SDK wrapper; and provide, from the merchant system via the SDK core, a secure payment container request to a payment gateway system.

[0013] In some non-limiting embodiments or aspects, the one or more processors are further programmed and/or configured to: receive, at the merchant system via the merchant webpage, account data associated with an account of a user, wherein the SDK wrapper is loaded in response to receiving the account data via the merchant webpage, and wherein the secure payment container request includes the account data.

[0014] In some non-limiting embodiments or aspects, the one or more processors are further programmed and/or configured to: receive, from the payment gateway system, a token associated with the account data in response to providing the secure payment container request.

[0015] In some non-limiting embodiments or aspects, determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core.

[0016] In some non-limiting embodiments or aspects, the secure payment container request is provided in response to a successful handshake between the SDK wrapper and the SDK core.

[0017] In some non-limiting embodiments or aspects, executing the SDK wrapper exposes the real function as a public method, and the real function throws an error in response to being called by the merchant system.

[0018] In some non-limiting embodiments or aspects, the SDK wrapper includes an Accept.js library, and the SDK core includes an AcceptCore.js library.

[0019] According to some non-limiting embodiments or aspects, provided is a computer program product including at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to: load, at a merchant system, a software development kit (SDK) wrapper from a payment gateway system via a merchant webpage associated with the merchant system; execute the SDK wrapper, wherein the SDK wrapper loads an SDK core when executed; determine an integrity of the SDK core; in response to determining the integrity of the SDK core, perform a handshake between the SDK wrapper and the SDK core and overloading a real function exported by the SDK wrapper; and provide, from the merchant system via the SDK core, a secure payment container request to a payment gateway system.

[0020] In some non-limiting embodiments or aspects, the instructions further cause the at least one processor to: receive, at the merchant system via the merchant webpage, account data associated with an account of a user, wherein the SDK wrapper is loaded in response to receiving the account data via the merchant webpage, and wherein the secure payment container request includes the account data.

[0021] In some non-limiting embodiments or aspects, the instructions further cause the at least one processor to: receive, from the payment gateway system, a token associated with the account data in response to providing the secure payment container request.

[0022] In some non-limiting embodiments or aspects, determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core.

[0023] In some non-limiting embodiments or aspects, the secure payment container request is provided in response to a successful handshake between the SDK wrapper and the SDK core.

[0024] In some non-limiting embodiments or aspects, executing the SDK wrapper exposes the real function as a public method, and the real function throws an error in response to being called by the merchant system.

[0025] In some non-limiting embodiments or aspects, the SDK wrapper includes an Accept.js library, and the SDK core includes an AcceptCore.js library.

[0026] Further embodiments or aspects are set forth in the following numbered clauses:

[0027] Clause 1. A computer-implemented method comprising: loading, with at least one processor, at a merchant system, a software development kit (SDK) wrapper from a payment gateway system via a merchant webpage associated with the merchant system; executing, with at least one processor, the SDK wrapper, wherein the SDK wrapper loads an SDK core when executed; determining, with at least one processor, an integrity of the SDK core; in response to determining the integrity of the SDK core, performing, with at least one processor, a handshake between the SDK wrapper and the SDK core and overloading a real function exported by the SDK wrapper; and providing, with at least one processor, from the merchant system via the SDK core, a secure payment container request to a payment gateway system.

[0028] Clause 2. The computer-implemented method of clause 1, further comprising: receiving, with at least one processor, at the merchant system via the merchant webpage, account data associated with an account of a user, wherein the SDK wrapper is loaded in response to receiving the account data via the merchant webpage, and wherein the secure payment container request includes the account data.

[0029] Clause 3. The computer-implemented method of clauses 1 or 2, further comprising: receiving, with at least one processor, from the payment gateway system, a token associated with the account data in response to providing the secure payment container request.

[0030] Clause 4. The computer-implemented method of any of clauses 1-3, wherein determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core.

[0031] Clause 5. The computer-implemented method of any of clauses 1-4, wherein the secure payment container request is provided in response to a successful handshake between the SDK wrapper and the SDK core.

[0032] Clause 6. The computer-implemented method of any of clauses 1-5, wherein executing the SDK wrapper exposes the real function as a public method, and wherein the real function throws an error in response to being called by the merchant system.

[0033] Clause 7. The computer-implemented method of any of clauses 1-6, wherein the SDK wrapper includes an Accept.js library, and wherein the SDK core includes an AcceptCore.js library.

[0034] Clause 8. A computing system comprising: one or more processors programmed and/or configured to: load, at a merchant system, a software development kit (SDK) wrapper from a payment gateway system via a merchant webpage associated with the merchant system; execute the SDK wrapper, wherein the SDK wrapper loads an SDK core when executed; determine an integrity of the SDK core; in response to determining the integrity of the SDK core, perform a handshake between the SDK wrapper and the SDK core and overloading a real function exported by the SDK wrapper; and provide, from the merchant system via the SDK core, a secure payment container request to a payment gateway system.

[0035] Clause 9. The computing system of clause 8, wherein the one or more processors are further programmed and/or configured to: receive, at the merchant system via the merchant webpage, account data associated with an account of a user, wherein the SDK wrapper is loaded in response to receiving the account data via the merchant webpage, and wherein the secure payment container request includes the account data.

[0036] Clause 10. The computing system of clauses 8 or 9, wherein the one or more processors are further programmed and/or configured to: receive, from the payment gateway system, a token associated with the account data in response to providing the secure payment container request.

[0037] Clause 11. The computing system of any of clauses 8-10, wherein determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core.

[0038] Clause 12. The computing system of any of clauses 8-11, wherein the secure payment container request is provided in response to a successful handshake between the SDK wrapper and the SDK core.

[0039] Clause 13. The computing system of any of clauses 8-12, wherein executing the SDK wrapper exposes the real function as a public method, and wherein the real function throws an error in response to being called by the merchant system.

[0040] Clause 14. The computing system of any of clauses 8-13, wherein the SDK wrapper includes an Accept.js library, and wherein the SDK core includes an AcceptCore.js library.

[0041] Clause 15. A computer program product comprising at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to: load, at a merchant system, a software development kit (SDK) wrapper from a payment gateway system via a merchant webpage associated with the merchant system; execute the SDK wrapper, wherein the SDK wrapper loads an SDK core when executed; determine an integrity of the SDK core; in response to determining the integrity of the SDK core, perform a handshake between the SDK wrapper and the SDK core and overloading a real function exported by the SDK wrapper; and provide, from the merchant system via the SDK core, a secure payment container request to a payment gateway system.

[0042] Clause 16. The computer program product of clause 15, wherein the instructions further cause the at least one processor to: receive, at the merchant system via the merchant webpage, account data associated with an account of a user, wherein the SDK wrapper is loaded in response to receiving the account data via the merchant webpage, and wherein the secure payment container request includes the account data.

[0043] Clause 17. The computer program product of clauses 15 or 16, wherein the instructions further cause the at least one processor to: receive, from the payment gateway system, a token associated with the account data in response to providing the secure payment container request.

[0044] Clause 18. The computer program product of any of clauses 15-17, wherein determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core.

[0045] Clause 19. The computer program product of any of clauses 15-18, wherein the secure payment container request is provided in response to a successful handshake between the SDK wrapper and the SDK core.

[0046] Clause 20. The computer program product of any of clauses 15-19, wherein executing the SDK wrapper exposes the real function as a public method, and wherein the real function throws an error in response to being called by the merchant system.

[0047] Clause 21. The computer program product of any of clauses 15-20, wherein the SDK wrapper includes an Accept.js library, and wherein the SDK core includes an AcceptCore.js library.

[0048] These and other features and characteristics of the present disclosure, as well as the methods of operation and functions of the related elements of structures and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only and are not intended as a definition of limits. As used in the specification and the claims, the singular form of “a,” “an,” and “the” include plural referents unless the context clearly dictates otherwise.

BRIEF DESCRIPTION OF THE DRAWINGS

[0049] Additional advantages and details are explained in greater detail below with reference to the exemplary embodiments or aspects that are illustrated in the accompanying schematic figures, in which:

[0050] FIG. 1 is a diagram of non-limiting embodiments or aspects of an environment in which systems, devices, products, apparatus, and/or methods, described herein, may be implemented;

(e.g., via communication network 114, etc.) and/or communicating information and/or data to merchant system 102, payment gateway system 104, acquirer system 106, issuer system 110, and/or user device 112 (e.g., via communication network 114, etc.). For example, transaction service provider system 108 may include a computing device, such as a server (e.g., a transaction processing server, etc.), a group of servers, and/or other like devices. In some non-limiting embodiments or aspects, transaction service provider system 108 may be associated with a transaction service provider as described herein. In some non-limiting embodiments or aspects, transaction service provider 108 may include and/or access one or more internal and/or external databases including account data, transaction data, merchant data, demographic data, and/or the like.

[0078] Issuer system 110 may include one or more devices capable of receiving information and/or data from merchant system 102, payment gateway system 104, acquirer system 106, transaction service provider system 108, and/or user device 112 (e.g., via communication network 114, etc.) and/or communicating information and/or data to merchant system 102, payment gateway system 104, acquirer system 106, transaction service provider system 108, and/or user device 112 (e.g., via communication network 114, etc.). For example, issuer system 110 may include a computing device, such as a server, a group of servers, and/or other like devices. In some non-limiting embodiments or aspects, issuer system 110 may be associated with an issuer institution as described herein. For example, issuer system 110 may be associated with an issuer institution that issued a payment account or instrument (e.g., a credit account, a debit account, a credit card, a debit card, etc.) to a user (e.g., a user associated with user device 112, etc.).

[0079] In some non-limiting embodiments or aspects, transaction processing network 101 includes a plurality of systems in a communication path for processing a transaction. For example, transaction processing network 101 may include merchant system 102, payment gateway system 104, acquirer system 106, transaction service provider system 108, and/or issuer system 110 in a communication path (e.g., a communication path, a communication channel, a communication network, etc.) for processing an electronic payment transaction. As an example, transaction processing network 101 may process

be loaded from payment gateway system 104 in response to receiving the account data via the merchant webpage.

[0095] In some non-limiting embodiments or aspects, in response to loading the Accept.js library, the Accept.js library exports a real function (e.g., a tokenization function, a secure payment container request function, etc.) as a public method, and if the public method is called by merchant system 102, the public method throws an error. Accordingly, merchant system 102 may be forced to implement the AcceptCore.js library, which is automatically loaded by the Accept.js library as described in more detail herein.

[0096] As shown in FIG. 3, at step 306, process 300 incudes executing an SDK wrapper to load an SDK core. For example, merchant system 102 may execute an SDK wrapper to load an SDK core. As an example, merchant system 102 may execute the SDK wrapper, wherein the SDK wrapper loads an SDK core (e.g., an AcceptCore.js library, etc.). In such an example, and referring again to FIG. 4, at step S406, the AcceptCore.js library may be automatically loaded from payment gateway system 104 (e.g., from a content delivery network (CDN) of Authorize.net, etc.) in response to merchant system 102 loading and/or executing the Accept.js library.

[0097] As shown in FIG. 3, at step 308, process 300 includes determining an integrity of an SDK core. For example, merchant system 102 may determine an integrity of an SDK core. As an example, merchant system 102 may determine the integrity of the SDK core. In such an example, and referring again to FIG. 4, at step S408, the Accept.js library automatically checks, in response to the AcceptCore.js library being loaded from payment gateway system 104, whether the integrity of the AcceptCore.js library matches a subresource integrity. In some non-limiting embodiments or aspects, determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core. For example, the Accept.js library may perform a W3C Subresource Integrity check hash on the AcceptCore.js library.

[0098] As shown in FIG. 3, at step 310, process 300 includes performing a handshake between an SDK wrapper and an SDK core. For example, merchant system 102 may perform a handshake between an SDK wrapper and an SDK core. As an example, in response to determining the integrity of the SDK core, merchant system 102 may perform a handshake between the SDK wrapper and the SDK core and overload a real function

WHAT IS CLAIMED IS:

1. A computer-implemented method comprising:

loading, with at least one processor, at a merchant system, a software development kit (SDK) wrapper from a payment gateway system via a merchant webpage associated with the merchant system;

executing, with at least one processor, the SDK wrapper, wherein the SDK wrapper loads an SDK core when executed;

determining, with at least one processor, an integrity of the SDK core;

in response to determining the integrity of the SDK core, performing, with at least one processor, a handshake between the SDK wrapper and the SDK core and overloading a real function exported by the SDK wrapper; and

providing, with at least one processor, from the merchant system via the SDK core, a secure payment container request to a payment gateway system.

2. The computer-implemented method of claim 1, further comprising:

receiving, with at least one processor, at the merchant system via the merchant webpage, account data associated with an account of a user, wherein the SDK wrapper is loaded in response to receiving the account data via the merchant webpage, and wherein the secure payment container request includes the account data.

3. The computer-implemented method of claim 1, further comprising:

receiving, with at least one processor, from the payment gateway system, a token associated with the account data in response to providing the secure payment container request.

4. The computer-implemented method of claim 1, wherein determining the integrity of the SDK core includes performing a W3C Subresource iIntegrity check hash on the SDK core.

10. The computing system of claim 8, wherein the one or more processors are further programmed and/or configured to:

receive, from the payment gateway system, a token associated with the account data in response to providing the secure payment container request.

11. The computing system of claim 8, wherein determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core.

12. The computing system of claim 8, wherein the secure payment container request is provided in response to a successful handshake between the SDK wrapper and the SDK core.

13. The computing system of claim 8, wherein executing the SDK wrapper exposes the real function as a public method, and wherein the real function throws an error in response to being called by the merchant system.

14. The computing system of claim 8, wherein the SDK wrapper includes an Accept.js library, and wherein the SDK core includes an AcceptCore.js library.

15. A computer program product comprising at least one non-transitory computer-readable medium including program instructions that, when executed by at least one processor, cause the at least one processor to:

load, at a merchant system, a software development kit (SDK) wrapper from a payment gateway system via a merchant webpage associated with the merchant system;

execute the SDK wrapper, wherein the SDK wrapper loads an SDK core when executed;

determine an integrity of the SDK core;

in response to determining the integrity of the SDK core, perform a handshake between the SDK wrapper and the SDK core and overloading a real function exported by the SDK wrapper; and

provide, from the merchant system via the SDK core, a secure payment container request to a payment gateway system.

16. The computer program product of claim 15, wherein the instructions further cause the at least one processor to:

receive, at the merchant system via the merchant webpage, account data associated with an account of a user, wherein the SDK wrapper is loaded in response to receiving the account data via the merchant webpage, and wherein the secure payment container request includes the account data.

17. The computer program product of claim 15, wherein the instructions further cause the at least one processor to:

receive, from the payment gateway system, a token associated with the account data in response to providing the secure payment container request.

18. The computer program product of claim 15, wherein determining the integrity of the SDK core includes performing a W3C Subresource Integrity check hash on the SDK core.

19. The computer program product of claim 15, wherein the secure payment container request is provided in response to a successful handshake between the SDK wrapper and the SDK core.

20. The computer program product of claim 15, wherein executing the SDK wrapper exposes the real function as a public method, and wherein the real function throws an error in response to being called by the merchant system.