





























































































instance fingerprint matching, because of a policy decision that such a technique is most suited to authentication for the particular vendor's purposes.

Such varying weights may be defined by the authentication requestor in generating the authentication request and sent to the trust engine 110 with the authentication request in one mode of operation. Such options could also be set as preferences during an initial enrollment process for the authentication requestor and stored within the authentication engine in another mode of operation.

Once the authentication engine 215 produces an authentication confidence level for the authentication data provided, this confidence level is used to complete the authentication request in STEP 1640, and this information is forwarded from the authentication engine 215 to the transaction engine 205 for inclusion in a message to the authentication requestor.

The process described above is merely exemplary, and those of skill in the art will recognize that the steps need not be performed in the order shown. Furthermore, certain steps, such as the evaluation of the reliability of each authentication instance provided, may be carried out in parallel with one another if circumstances permit.

In a further aspect of this invention, a method is provided to accommodate conditions when the authentication confidence level produced by the process described above fails to meet the required trust level of the vendor or other party requiring the authentication. In circumstances such as these where a gap exists between the level of confidence provided and the level of trust desired, the operator of the trust engine 110 is in a position to provide opportunities for one or both parties to provide alternate data or requirements in order to close this trust gap. This process will be referred to as "trust arbitrage" herein.

Trust arbitrage may take place within a framework of cryptographic authentication as described above with reference to FIGURES 10 and 11. As shown therein, a vendor or other party will request authentication of a particular user in association with a particular transaction. In one circumstance, the vendor simply requests an authentication, either positive or negative, and after receiving appropriate data from the user, the trust engine 110 will provide such a binary authentication. In circumstances such as these, the degree of confidence required in order to secure a positive authentication is determined based upon preferences set within the trust engine 110.

However, it is also possible that the vendor may request a particular level of trust in order to complete a particular transaction. This required level may be included with the authentication request (e.g. authenticate this user to 96% confidence) or may be determined by the trust engine 110 based on other factors associated with the transaction (i.e. authenticate this user as appropriate for this transaction). One such factor might be the economic value of the transaction. For transactions which have greater economic value, a higher degree of trust may be required. Similarly, for transactions with high degrees of risk a high degree of trust may be required. Conversely, for transactions which are either of low risk or of low value, lower trust levels may be required by the vendor or other authentication requestor.

The process of trust arbitrage occurs between the steps of the trust engine 110 receiving the authentication data in STEP 1050 of FIGURE 10 and the return of an authentication result to the vendor in STEP 1055 of FIGURE 10.

Between these steps, the process which leads to the evaluation of trust levels and the potential trust arbitrage occurs as shown in FIGURE 17. In circumstances where simple binary authentication is performed, the process shown in FIGURE 17 reduces to having the transaction engine 205 directly compare the authentication data provided with the enrollment data for the identified user as discussed above with reference to FIGURE 10, flagging any difference as a negative authentication.

As shown in FIGURE 17, the first step after receiving the data in STEP 1050 is for the transaction engine 205 to determine the trust level which is required for a positive authentication for this particular transaction in STEP 1710. This step may be performed by one of several different methods. The required trust level may be specified to the trust engine 110 by the authentication requestor at the time when the authentication request is made. The authentication requestor may also set a preference in advance which is stored within the depository 210 or other storage which is accessible by the transaction engine 205. This preference may then be read and used each time an authentication request is made by this authentication requestor. The preference may also be associated with a particular user as a security measure such that a particular level of trust is always required in order to authenticate that user, the user preference being stored in the depository 210 or other storage media accessible by the transaction engine 205. The required level may also be derived by the transaction engine 205 or authentication engine 215 based upon information provided in the authentication request, such as the value and risk level of the transaction to be authenticated.

In one mode of operation, a policy management module or other software which is used when generating the authentication request is used to specify the required degree of trust for the authentication of the transaction. This may be used to provide a series of rules to follow when assigning the required level of trust based upon the policies which are specified within the policy management module. One advantageous mode of operation is for such a module to be incorporated with the web server of a vendor in order to appropriately determine required level of trust for transactions initiated with the vendor's web server. In this way, transaction requests from users may be assigned a required trust level in accordance with the policies of the vendor and such information may be forwarded to the trust engine 110 along with the authentication request.

This required trust level correlates with the degree of certainty that the vendor wants to have that the individual authenticating is in fact who he identifies himself as. For example, if the transaction is one where the vendor wants a fair degree of certainty because goods are changing hands, the vendor may require a trust level of 85%. For situation where the vendor is merely authenticating the user to allow him to view members only content or exercise privileges on a chat room, the downside risk may be small enough that the vendor requires only a 60% trust level. However, to enter into a production contract with a value of tens of thousands of dollars, the vendor may require a trust level of 99% or more.

This required trust level represents a metric to which the user must authenticate himself in order to complete the transaction. If the required trust level is 85% for example, the user must provide authentication to the trust engine 110 sufficient for the trust engine 110 to say with 85% confidence that the user is who they say they are. It is the

balance between this required trust level and the authentication confidence level which produces either a positive authentication (to the satisfaction of the vendor) or a possibility of trust arbitrage.

As shown in FIGURE 17, after the transaction engine 205 receives the required trust level, it compares in STEP 1720 the required trust level to the authentication confidence level which the authentication engine 215 calculated for the current authentication (as discussed with reference to FIGURE 16). If the authentication confidence level is higher than the required trust level for the transaction in STEP 1730, then the process moves to STEP 1740 where a positive authentication for this transaction is produced by the transaction engine 205. A message to this effect will then be inserted into the authentication results and returned to the vendor by the transaction engine 205 as shown in STEP 1055 (see FIGURE 10).

However, if the authentication confidence level does not fulfill the required trust level in STEP 1730, then a confidence gap exists for the current authentication, and trust arbitrage is conducted in STEP 1750. Trust arbitrage is described more completely with reference to FIGURE 18 below. This process as described below takes place within the transaction engine 205 of the trust engine 110. Because no authentication or other cryptographic operations are needed to execute trust arbitrage (other than those required for the SSL communication between the transaction engine 205 and other components), the process may be performed outside the authentication engine 215. However, as will be discussed below, any reevaluation of authentication data or other cryptographic or authentication events will require the transaction engine 205 to resubmit the appropriate data to the authentication engine 215. Those of skill in the art will recognize that the trust arbitrage process could alternately be structured to take place partially or entirely within the authentication engine 215 itself.

As mentioned above, trust arbitrage is a process where the trust engine 110 mediates a negotiation between the vendor and user in an attempt to secure a positive authentication where appropriate. As shown in STEP 1805, the transaction engine 205 first determines whether or not the current situation is appropriate for trust arbitrage. This may be determined based upon the circumstances of the authentication, e.g. whether this authentication has already been through multiple cycles of arbitrage, as well as upon the preferences of either the vendor or user, as will be discussed further below.

In such circumstances where arbitrage is not possible, the process proceeds to STEP 1810 where the transaction engine 205 generates a negative authentication and then inserts it into the authentication results which are sent to the vendor in STEP 1055 (see FIGURE 10). One limit which may be advantageously used to prevent authentications from pending indefinitely is to set a time-out period from the initial authentication request. In this way, any transaction which is not positively authenticated within the time limit is denied further arbitrage and negatively authenticated. Those of skill in the art will recognize that such a time limit may vary depending upon the circumstances of the transaction and the desires of the user and vendor. Limitations may also be placed upon the number of attempts that may be made at providing a successful authentication. Such limitations may be handled by an attempt limiter 535 as shown in FIGURE 5.

If arbitrage is not prohibited in STEP 1805, the transaction engine 205 will then engage in negotiation with one or both of the transacting parties. The transaction engine 205 may send a message to the user requesting some form of additional authentication in order to boost the authentication confidence level produced as shown in STEP 1820. In the simplest form, this may simply indicate that authentication was insufficient. A request to produce one or more additional authentication instances to improve the overall confidence level of the authentication may also be sent.

If the user provides some additional authentication instances in STEP 1825, then the transaction engine 205 adds these authentication instances to the authentication data for the transaction and forwards it to the authentication engine 215 as shown in STEP 1015 (see FIGURE 10), and the authentication is reevaluated based upon both the pre-existing authentication instances for this transaction and the newly provided authentication instances.

An additional type of authentication may be a request from the trust engine 110 to make some form of person-to-person contact between the trust engine 110 operator (or a trusted associate) and the user, for example, by phone call. This phone call or other non-computer authentication can be used to provide personal contact with the individual and also to conduct some form of questionnaire based authentication. This also may give the opportunity to verify an originating telephone number and potentially a voice analysis of the user when he calls in. Even if no additional authentication data can be provided, the additional context associated with the user's phone number may improve the reliability of the authentication context. Any revised data or circumstances based upon this phone call are fed into the trust engine 110 for use in consideration of the authentication request.

Additionally, in STEP 1820 the trust engine 110 may provide an opportunity for the user to purchase insurance, effectively buying a more confident authentication. The operator of the trust engine 110 may, at times, only want to make such an option available if the confidence level of the authentication is above a certain threshold to begin with. In effect, this user side insurance is a way for the trust engine 110 to vouch for the user when the authentication meets the normal required trust level of the trust engine 110 for authentication, but does not meet the required trust level of the vendor for this transaction. In this way, the user may still successfully authenticate to a very high level as may be required by the vendor, even though he only has authentication instances which produce confidence sufficient for the trust engine 110.

This function of the trust engine 110 allows the trust engine 110 to vouch for someone who is authenticated to the satisfaction of the trust engine 110, but not of the vendor. This is analogous to the function performed by a notary in adding his signature to a document in order to indicate to someone reading the document at a later time that the person whose signature appears on the document is in fact the person who signed it. The signature of the notary testifies to the act of signing by the user. In the same way, the trust engine is providing an indication that the person transacting is who they say they are.

However, because the trust engine 110 is artificially boosting the level of confidence provided by the user, there is a greater risk to the trust engine 110 operator, since the user is not actually meeting the required trust level of the vendor. The cost of the insurance is designed to offset the risk of a false positive authentication to the trust

engine 110 (who may be effectively notarizing the authentications of the user). The user pays the trust engine 110 operator to take the risk of authenticating to a higher level of confidence than has actually been provided.

5 Because such an insurance system allows someone to effectively buy a higher confidence rating from the trust engine 110, both vendors and users may wish to prevent the use of user side insurance in certain transactions. Vendors may wish to limit positive authentications to circumstances where they know that actual authentication data supports the degree of confidence which they require and so may indicate to the trust engine 110 that user side insurance is not to be allowed. Similarly, to protect his online identity, a user may wish to prevent the use of user side insurance on his account, or may wish to limit its use to situations where the authentication confidence level without the insurance is higher than a certain limit. This may be used as a security measure to prevent someone from  
10 overhearing a password or stealing a smart card and using them to falsely authenticate to a low level of confidence, and then purchasing insurance to produce a very high level of (false) confidence. These factors may be evaluated in determining whether user side insurance is allowed.

If user purchases insurance in STEP 1840, then the authentication confidence level is adjusted based upon the insurance purchased in STEP 1845, and the authentication confidence level and required trust level are again  
15 compared in STEP 1730 (see FIGURE 17). The process continues from there, and may lead to either a positive authentication in STEP 1740 (see FIGURE 17), or back into the trust arbitrage process in STEP 1750 for either further arbitrage (if allowed) or a negative authentication in STEP 1810 if further arbitrage is prohibited.

In addition to sending a message to the user in STEP 1820, the transaction engine 205 may also send a message to the vendor in STEP 1830 which indicates that a pending authentication is currently below the required  
20 trust level. The message may also offer various options on how to proceed to the vendor. One of these options is to simply inform the vendor of what the current authentication confidence level is and ask if the vendor wishes to maintain their current unfulfilled required trust level. This may be beneficial because in some cases, the vendor may have independent means for authenticating the transaction or may have been using a default set of requirements which generally result in a higher required level being initially specified than is actually needed for the particular transaction  
25 at hand.

For instance, it may be standard practice that all incoming purchase order transactions with the vendor are expected to meet a 98% trust level. However, if an order was recently discussed by phone between the vendor and a long-standing customer, and immediately thereafter the transaction is authenticated, but only to a 93% confidence  
30 level, the vendor may wish to simply lower the acceptance threshold for this transaction, because the phone call effectively provides additional authentication to the vendor. In certain circumstances, the vendor may be willing to lower their required trust level, but not all the way to the level of the current authentication confidence. For instance, the vendor in the above example might consider that the phone call prior to the order might merit a 4% reduction in the degree of trust needed; however, this is still greater than the 93% confidence produced by the user.

If the vendor does adjust their required trust level in STEP 1835, then the authentication confidence level  
35 produced by the authentication and the required trust level are compared in STEP 1730 (see FIGURE 17). If the

confidence level now exceeds the required trust level, a positive authentication may be generated in the transaction engine 205 in STEP 1740 (see FIGURE 17). If not, further arbitrage may be attempted as discussed above if it is permitted.

5 In addition to requesting an adjustment to the required trust level, the transaction engine 205 may also offer vendor side insurance to the vendor requesting the authentication. This insurance serves a similar purpose to that described above for the user side insurance. Here, however, rather than the cost corresponding to the risk being taken by the trust engine 110 in authenticating above the actual authentication confidence level produced, the cost of the insurance corresponds to the risk being taken by the vendor in accepting a lower trust level in the authentication.

10 Instead of just lowering their actual required trust level, the vendor has the option of purchasing insurance to protect itself from the additional risk associated with a lower level of trust in the authentication of the user. As described above, it may be advantageous for the vendor to only consider purchasing such insurance to cover the trust gap in conditions where the existing authentication is already above a certain threshold.

15 The availability of such vendor side insurance allows the vendor the option to either: lower his trust requirement directly at no additional cost to himself, bearing the risk of a false authentication himself (based on the lower trust level required); or, buying insurance for the trust gap between the authentication confidence level and his requirement, with the trust engine 110 operator bearing the risk of the lower confidence level which has been provided. By purchasing the insurance, the vendor effectively keeps his high trust level requirement, because the risk of a false authentication is shifted to the trust engine 110 operator.

20 If the vendor purchases insurance in STEP 1840, the authentication confidence level and required trust levels are compared in STEP 1730 (see FIGURE 17), and the process continues as described above.

25 Note that it is also possible that both the user and the vendor respond to messages from the trust engine 110. Those of skill in the art will recognize that there are multiple ways in which such situations can be handled. One advantageous mode of handling the possibility of multiple responses is simply to treat the responses in a first-come, first-served manner. For example, if the vendor responds with a lowered required trust level and immediately thereafter the user also purchases insurance to raise his authentication level, the authentication is first reevaluated based upon the lowered trust requirement from the vendor. If the authentication is now positive, the user's insurance purchase is ignored. In another advantageous mode of operation, the user might only be charged for the level of insurance required to meet the new, lowered trust requirement of the vendor (if a trust gap remained even with the lowered vendor trust requirement).

30 If no response from either party is received during the trust arbitrage process at STEP 1850 within the time limit set for the authentication, the arbitrage is reevaluated in STEP 1805. This effectively begins the arbitrage process again. If the time limit was final or other circumstances prevent further arbitrage in STEP 1805, a negative authentication is generated by the transaction engine 205 in STEP 1810 and returned to the vendor in STEP 1055 (see FIGURE 10). If not, new messages may be sent to the user and vendor, and the process may be repeated as desired.

Note that for certain types of transactions, for instance, digitally signing documents which are not part of a transaction, there may not necessarily be a vendor or other third party; therefore the transaction is primarily between the user and the trust engine 110. In circumstances such as these, the trust engine 110 will have its own required trust level which must be satisfied in order to generate a positive authentication. However, in such circumstances, it will often not be desirable for the trust engine 110 to offer insurance to the user in order for him to raise the confidence of his own signature.

The process described above and shown in FIGURES 16 - 18 may be carried out using various communications modes as described above with reference to the trust engine 110. For instance, the messages may be web-based and sent using SSL connections between the trust engine 110 and applets downloaded in real time to browsers running on the user or vendor systems. In an alternate mode of operation, certain dedicated applications may be in use by the user and vendor which facilitate such arbitrage and insurance transactions. In another alternate mode of operation, secure email operations may be used to mediate the arbitrage described above, thereby allowing deferred evaluations and batch processing of authentications. Those of skill in the art will recognize that different communications modes may be used as are appropriate for the circumstances and authentication requirements of the vendor.

The following description with reference to FIGURE 19 describes a sample transaction which integrates the various aspects of the present invention as described above. This example illustrates the overall process between a user and a vendor as mediated by the trust engine 110. Although the various steps and components as described in detail above may be used to carry out the following transaction, the process illustrated focuses on the interaction between the trust engine 110, user and vendor.

The transaction begins when the user, while viewing web pages online, fills out an order form on the web site of the vendor in STEP 1900. The user wishes to submit this order form to the vendor, signed with his digital signature. In order to do this, the user submits the order form with his request for a signature to the trust engine 110 in STEP 1905. The user will also provide authentication data which will be used as described above to authenticate his identity.

In STEP 1910 the authentication data is compared to the enrollment data by the trust engine 110 as discussed above, and if a positive authentication is produced, the hash of the order form, signed with the private key of the user, is forwarded to the vendor along with the order form itself.

The vendor receives the signed form in STEP 1915, and then the vendor will generate an invoice or other contract related to the purchase to be made in STEP 1920. This contract is sent back to the user with a request for a signature in STEP 1925. The vendor also sends an authentication request for this contract transaction to the trust engine 110 in STEP 1930 including a hash of the contract which will be signed by both parties. To allow the contract to be digitally signed by both parties, the vendor also includes authentication data for itself so that the vendor's signature upon the contract can later be verified if necessary.

As discussed above, the trust engine 110 then verifies the authentication data provided by the vendor to confirm the vendor's identity, and if the data produces a positive authentication in STEP 1935, continues with STEP 1955 when the data is received from the user. If the vendor's authentication data does not match the enrollment data of the vendor to the desired degree, a message is returned to the vendor requesting further authentication. Trust  
5 arbitrage may be performed here if necessary, as described above, in order for the vendor to successfully authenticate itself to the trust engine 110.

When the user receives the contract in STEP 1940, he reviews it, generates authentication data to sign it if it is acceptable in STEP 1945, and then sends a hash of the contract and his authentication data to the trust engine 110 in STEP 1950. The trust engine 110 verifies the authentication data in STEP 1955 and if the authentication is  
10 good, proceeds to process the contract as described below. As discussed above with reference to FIGURES 17 and 18, trust arbitrage may be performed as appropriate to close any trust gap which exists between the authentication confidence level and the required authentication level for the transaction.

The trust engine 110 signs the hash of the contract with the user's private key, and sends this signed hash to the vendor in STEP 1960, signing the complete message on its own behalf, i.e. including a hash of the complete  
15 message (including the user's signature) encrypted with the private key 510 of the trust engine 110. This message is received by the vendor in STEP 1965. The message represents a signed contract (hash of contract encrypted using user's private key) and a receipt from the trust engine 110 (the hash of the message including the signed contract, encrypted using the trust engine 110's private key).

The trust engine 110 similarly prepares a hash of the contract with the vendor's private key in STEP 1970,  
20 and forwards this to the user, signed by the trust engine 110. In this way, the user also receives a copy of the contract, signed by the vendor, as well as a receipt, signed by the trust engine 110, for delivery of the signed contract in STEP 1975.

In addition to the foregoing, an additional aspect of the invention provides a cryptographic Service Provider Module (SPM) which may be available to a client side application as a means to access functions provided by the trust  
25 engine 110 described above. One advantageous way to provide such a service is for the cryptographic SPM is to mediate communications between a third party Application Programming Interface (API) and a trust engine 110 which is accessible via a network or other remote connection. A sample cryptographic SPM is described below with reference to FIGURE 20.

For example, on a typical system, a number of API's are available to programmers. Each API provides a set  
30 of function calls which may be made by an application 2000 running upon the system. Examples of API's which provide programming interfaces suitable for cryptographic functions, authentication functions, and other security function include the Cryptographic API (CAPI) 2010 provided by Microsoft with its Windows operating systems, and the Common Data Security Architecture (CDSA), sponsored by IBM, Intel and other members of the Open Group. CAPI will be used as an exemplary security API in the discussion that follows. However, the cryptographic SPM described  
35 could be used with CDSA or other security API's as are known in the art.

This API is used by a user system 105 or vendor system 120 when a call is made for a cryptographic function. Included among these functions may be requests associated with performing various cryptographic operations, such as encrypting a document with a particular key, signing a document, requesting a digital certificate, verifying a signature upon a signed document, and such other cryptographic functions as are described above or known to those of skill in the art.

Such cryptographic functions are normally performed locally to the system upon which CAPI 2010 is located. This is because generally the functions called require the use of either resources of the local user system 105, such as a fingerprint reader, or software functions which are programmed using libraries which are executed on the local machine. Access to these local resources is normally provided by one or more Service Provider Modules (SPM's) 2015, 2020 as referred to above which provide resources with which the cryptographic functions are carried out. Such SPM's may include software libraries 2015 to perform encrypting or decrypting operations, or drivers and applications 2020 which are capable of accessing specialized hardware 2025, such as biometric scanning devices. In much the way that CAPI 2010 provides functions which may be used by an application 2000 of the system 105, the SPM's 2015, 2020 provide CAPI with access to the lower level functions and resources associated with the available services upon the system.

In accordance with the invention, it is possible to provide a cryptographic SPM 2030 which is capable of accessing the cryptographic functions provided by the trust engine 110 and making these functions available to an application 2000 through CAPI 2010. Unlike embodiments where CAPI 2010 is only able to access resources which are locally available through SPM's 2015, 2020, a cryptographic SPM 2030 as described herein would be able to submit requests for cryptographic operations to a remotely-located, network-accessible trust engine 110 in order to perform the operations desired.

For instance, if an application 2000 has a need for a cryptographic operation, such as signing a document, the application 2000 makes a function call to the appropriate CAPI 2010 function. CAPI 2010 in turn will execute this function, making use of the resources which are made available to it by the SPM's 2015, 2020 and the cryptographic SPM 2030. In the case of a digital signature function, the cryptographic SPM 2030 will generate an appropriate request which will be sent to the trust engine 110 across the communication link 125.

The operations which occur between the cryptographic SPM 2030 and the trust engine 110 are the same operations that would be possible between any other system and the trust engine 110. However, these functions are effectively made available to a user system 105 through CAPI 2010 such that they appear to be locally available upon the user system 105 itself. However, unlike ordinary SPM's 2015, 2020, the functions are being carried out on the remote trust engine 110 and the results relayed to the cryptographic SPM 2030 in response to appropriate requests across the communication link 125.

This cryptographic SPM 2030 makes a number of operations available to the user system 105 or a vendor system 120 which might not otherwise be available. These functions include without limitation: encryption and

decryption of documents; issuance of digital certificates; digital signing of documents; verification of digital signatures; and such other operations as will be apparent to those of skill in the art.

5           Additionally, other combinations, admissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Accordingly, the present invention is not intended to be limited by the reaction of the preferred embodiments but is to be defined by a reference to the appended claims.

**WHAT IS CLAIMED IS:**

1. A method of implementing an authentication transaction related to an electronic transaction between a vendor and a user, the method comprising:

5 receiving a request for an authentication transaction from a vendor, wherein the authentication transaction is related to an electronic transaction involving the vendor and a user, and the authentication transaction includes a transaction identifier (TID);

forwarding enrollment authentication data corresponding to the user and the TID to an authentication engine;

10 querying the user for current authentication data and the TID;

receiving the current authentication data along with the TID from the user and forwarding the current authentication data and the TID to the authentication engine;

matching the enrollment authentication data to the current authentication data through the TID; and

15 comparing the enrollment authentication data to the current authentication data in order to generate an authentication result.

2. A method of implementing a cryptographic transaction related to an electronic transaction between a vendor and a user, the method comprising:

20 associating a user from multiple users with one or more keys from a plurality of private cryptographic keys stored on a secure server, wherein the one or more keys are unknown to the user;

receiving a request for a cryptographic transaction from a vendor, wherein the cryptographic transaction is related to an electronic transaction involving the vendor and the user and the cryptographic transaction includes a transaction identifier (TID);

25 forwarding enrollment authentication data corresponding to the user and the TID to an authentication engine;

querying the user for current authentication data and the TID;

receiving the current authentication data along with the TID from the user and forwarding the current authentication data and the TID to the authentication engine;

30 matching the enrollment authentication data to the current authentication data through the TID;

comparing the enrollment authentication data to the current authentication data in order to generate an authentication result; and

when the authentication result uniquely identifies the user, employing the one or more keys to perform one or more cryptographic functions.

35

3. The method of Claim 2, wherein the one or more cryptographic functions include one of digital signing, encryption, decryption, hash creation, key generation, and key destruction.

4. The method of Claim 2 wherein the electronic transaction represents logging into a portal which provides access to one or more services which require authentication of the user.

5. The method of Claim 4 additionally comprising unlocking a password vault associated with the user, the password vault providing access to additional data which is used to authenticate the user to additional services.

6. The method of Claim 5 wherein the additional data comprises passwords associated with the user.

7. The method of Claim 5 wherein the additional data comprises private cryptographic keys associated with the user.

8. The method of Claim 5 wherein the vendor is a trust engine and the password vault is associated with the trust engine.

9. The method of Claim 5 wherein the vendor is the operator of the portal and the password vault is associated with the vendor.

10. The method of Claim 5 wherein the password vault is stored on a system associated with the user.

11. A method of facilitating a cryptographic function related to an electronic transaction, the method comprising:

receiving at a trust engine, data corresponding to a transaction between at least a user and a vendor;

receiving at the trust engine, a request from the vendor to obtain the digital signature of the user;

and

generating the digital signature at the trust engine by using at least one private cryptographic key associated with the user.

12. The method of Claim 11, wherein the step of generating the digital signature further comprises acquiring an appropriate certificate corresponding to the at least one private cryptographic key.

13. A cryptographic system for performing cryptographic functions, the cryptographic system comprising:

data corresponding to a transaction between at least a user and a vendor;

a request from the vendor to obtain a digital signature of the user; and  
a cryptographic handling module which receives the data and the request, and which generates the digital signature by using at least one private cryptographic key associated with the user.

5           14.    A method of facilitating a cryptographic function related to an electronic transaction, the method comprising:

receiving at a first entity, data about a transaction between at least a second entity and a third entity;

receiving at the first entity, a request from the third entity to obtain the digital signature of the  
10           second entity; and

signing at the first entity the transaction data using at least one private cryptographic key associated with the second entity.

15           15.    A method of facilitating a cryptographic function related to an electronic transaction, the method comprising:

associating in a trust engine, a user from multiple users with one or more keys from a plurality of private cryptographic keys stored on the trust engine;

receiving at a trust engine, transaction data related to a transaction between at least a user and a vendor;

receiving at the trust engine, a request to digitally sign the transaction data; and  
20           signing the transaction data with the one or more keys associated with the user.

25           16.    A cryptographic system for performing cryptographic functions, the cryptographic system comprising:

a computer readable storage medium storing a plurality of private cryptographic keys;

transaction data related to a transaction between at least a user and a vendor; and

a cryptographic handling module which associates the user with one or more keys of the plurality of private cryptographic keys, which receives the request and the transaction data, and which signs the transaction data with the one or more keys associated with the user.

30           17.    A method of facilitating a cryptographic function related to an electronic transaction, the method comprising:

receiving transaction data that comprises a portion of an agreement between at least a first and second entity;

receiving a request for a cryptographic function; and

35           signing the transaction data with at least one private cryptographic key stored on a secure server.

18. A method of facilitating a cryptographic transaction related to an electronic transaction between first and second users, the method comprising:
- receiving authentication data from a first user;
  - 5 receiving transaction data identifying a cryptographic transaction between the first user and a second user;
  - comparing the authentication data to enrollment authentication data corresponding to the first user, thereby verifying the identity of the first user;
  - utilizing at least one private cryptographic key stored on a secure server and unknown to either user to perform cryptographic functions relating to the cryptographic transaction without releasing the at least one private cryptographic key to either user; and
  - 10 upon authentication of the identity of at least the first user, transmitting data related to the cryptographic transaction to one of the first and second users, without including additional cryptographic keys.
19. The method of Claim 18, wherein the at least one private cryptographic key is associated with the secure server.
20. The method of Claim 18, wherein the at least one private cryptographic key is associated with one of the first or second users.
21. The method of Claim 17, wherein the step of comparing the authentication data includes voiding the cryptographic transaction when a number of authentication attempts related to the cryptographic transaction reaches a predetermined amount.
22. A method as in Claim 18, wherein the authentication data comprises circumstantial data associated with the generation of the authentication data by the first user.
23. A method as in Claim 22, wherein a reliability is associated with the authentication based upon the comparison of the circumstantial data to circumstantial data acquired prior to the transaction on one or more occasions.
24. A method of increasing the speed of an authentication process by performing various authentication steps in parallel, the method comprising:
- 35 receiving a request for a cryptographic function related to an electronic transaction involving a vendor and a user;
  - processing the cryptographic function while awaiting authentication data from the user;

receiving the authentication data from the user; and  
transmitting response related to the cryptographic function when the authentication data from the user uniquely identifies the user.

5           25.    The method of Claim 24, wherein the step of processing further comprises retrieving enrollment authentication data.

          26.    The method of Claim 24, wherein the step of processing further comprises retrieving one or more cryptographic keys associated with one of the user and the vendor.

10

          27.    The method of Claim 26, wherein the step of retrieving one or more cryptographic keys further comprises obtaining an appropriate key type.

15

          28.    The method of Claim 24, wherein the step of performing the cryptographic function further comprises generating a digital signature for one of the user and the vendor.

          29.    A method of increasing the speed of an authentication process by performing various authentication steps in parallel, the method comprising:

20

          receiving an authentication request from a vendor wherein the authentication request is related to an electronic transaction involving the vendor and a user;

          retrieving enrollment data associated with the user while awaiting authentication data from the user; and

          receiving the authentication data from the user and comparing the authentication data to the retrieved enrollment data.

25

          30.    The method of Claim 29, wherein the authentication request comprises a user ID identifying the user.

30

          31.    The method of Claim 30, wherein the authentication request further comprises a transaction ID identifying the electronic transaction.

          32.    The method of Claim 24, wherein the authentication request comprises a transaction ID identifying the electronic transaction.

35

          33.    The method of Claim 32, wherein the transaction ID comprises a unique number.

          34.    The method of Claim 29, wherein the authentication request comprises a vendor ID identifying the vendor.

35. The method of Claim 29, wherein the authentication request comprises data indicating a type of authentication data requested from the user.

5 36. The method of Claim 35, wherein the type of authentication data comprises biometric data.

37. The method of Claim 29, wherein the step of retrieving enrollment data further comprises retrieving substantially randomized portions of enrollment data, each substantially randomized portion being retrieved from a different one of a plurality of data storage facilities.

10 38. The method of Claim 37, wherein the step of retrieving enrollment data further comprises assembling the substantially randomized portions of enrollment data into a deciphered usable form.

15 39. The method of Claim 29, wherein the step of comparing the authentication data to the enrollment data includes voiding the authentication request when a number of failed comparison attempts is greater than a predetermined threshold.

40. A method of conducting an authentication transaction to secure an electronic transaction, the method comprising:

20 receiving an authentication request associated with an electronic transaction from a first user;  
receiving data from a second user, the data comprising authentication data generated by the second user and circumstantial data associated with the generation of the authentication data;  
determining a level of trust associated with the authentication data, the level of trust based in part upon the circumstantial data; and  
25 providing a response to the authentication request to one of the first user and the second user indicating that the second user has been authenticated.

41. A method as in Claim 40 wherein the first user represents a seller of goods and the electronic transaction represents a sale of goods by the first user.

30 42. A method as in Claim 40 wherein the first user represents the operator of a server and the electronic transaction represents an access to the server by the first user.

43. A method as in Claim 40 wherein the first user represents a email provider and the electronic transaction represents signing an email message by the first user.

35 44. A method as in Claim 40 further comprising comparing the level of trust associated with the authentication data with a desired level of trust.

45. A method as in Claim 44 wherein the desired level of trust is determined at least in part by the properties of the transaction associated with the authentication request.
- 5 46. A method as in Claim 44 wherein the desired level of trust is specified by the first user in the authentication request.
47. A method for conducting an authentication transaction to provide security for an electronic transaction, the method comprising:
- 10 receiving authentication data from a first user, wherein the authentication data is associated with an authentication transaction;
- obtaining data corresponding to the circumstances surrounding the process of entering the authentication data;
- determining a reliability for the authentication data based upon the obtained data;
- 15 comparing the authentication data to previously stored enrollment data;
- generating a level of trust for the authentication data associated with the authentication transaction; and
- providing a response for use in the transaction.
- 20 48. A method as in Claim 47 wherein the authentication data is received over a computer network and the response is sent over the computer network.
49. A method as in Claim 48 wherein the authentication data is received via an email message and the response is sent via an email message.
- 25 50. A method as in Claim 48 wherein the authentication data is received from a client-side applet.
51. A method as in Claim 47 wherein the electronic transaction comprises cryptographically signing an email message by the first user.
- 30 52. A method as in Claim 47 wherein the electronic transaction comprises cryptographically signing a document by the first user.
53. A method as in Claim 52 further comprising encrypting a hash of the document with a private key associated with the first user.
- 35

54. A method as in Claim 47 wherein the electronic transaction comprises encrypting a document by the first user such that it is only readable by a third user.

55. A method as in Claim 47 wherein the electronic transaction comprises decrypting a document with a private key associated with the first user.

56. A system for authenticating an electronic transaction between a first user and a second user, comprising:

authentication data received from a first user and associated with an authentication transaction;

circumstantial data corresponding to the circumstances surrounding the authentication data;

reliability data associated with the authentication data and based upon the circumstantial data;

enrollment data associated with the first user; and

a trust engine which compares the authentication data to the enrollment data and generates a level of trust for the authentication data and provides the level of trust for use authenticating the first user.

57. A system as in Claim 56 wherein the trust engine compares the level of trust to a desired level of trust provided by the second user.

58. A system as in Claim 56 wherein the system further comprises a private key associated with the first user, and wherein the trust engine uses the private key to record the assent of the first user to the electronic transaction.

59. An apparatus that provides cryptographic functions by interconnecting a client-side application programming interface with a network-based trust engine, comprising:

a client-side application programming interface that is configured to operate on a client computer;

a service provider module that is in communication with the application programming interface and in communication with a network-based trust engine, wherein the service provider module is configured to request cryptographic functions from the network-based trust engine.

60. An apparatus as in Claim 59 wherein the cryptographic functions comprise securely authenticating a user to the client-side application programming interface.

61. An apparatus as in Claim 59 wherein the cryptographic functions comprise digitally signing a message for the client-side application programming interface.

62. An apparatus as in Claim 59 wherein the cryptographic functions comprise encrypting a message.

63. A method for performing security functions with a network-based cryptographic server, comprising:

receiving at a user system, a request for a security function from an application programming interface; and  
routing the request to a network-based cryptographic server.

5           64.    A method as in Claim 63 wherein the security function comprises an attempt to authenticate a user on the user system.

65.    A method as in Claim 63 wherein the security function comprises encrypting a message.

10       66.    A method as in Claim 63 wherein the security function comprises digitally signing a message.

67.    An apparatus for making security functions available to a user system from a network based cryptographic server, comprising:

15           a cryptographic service provider module in communication with an application programming interface of a user system, the cryptographic service provider module receiving requests from the applications programming interface of a user system; and

          a cryptographic server in communication with the cryptographic service provider module across a communication link, the cryptographic server providing responses to requests from the cryptographic service provider module across the communication link.

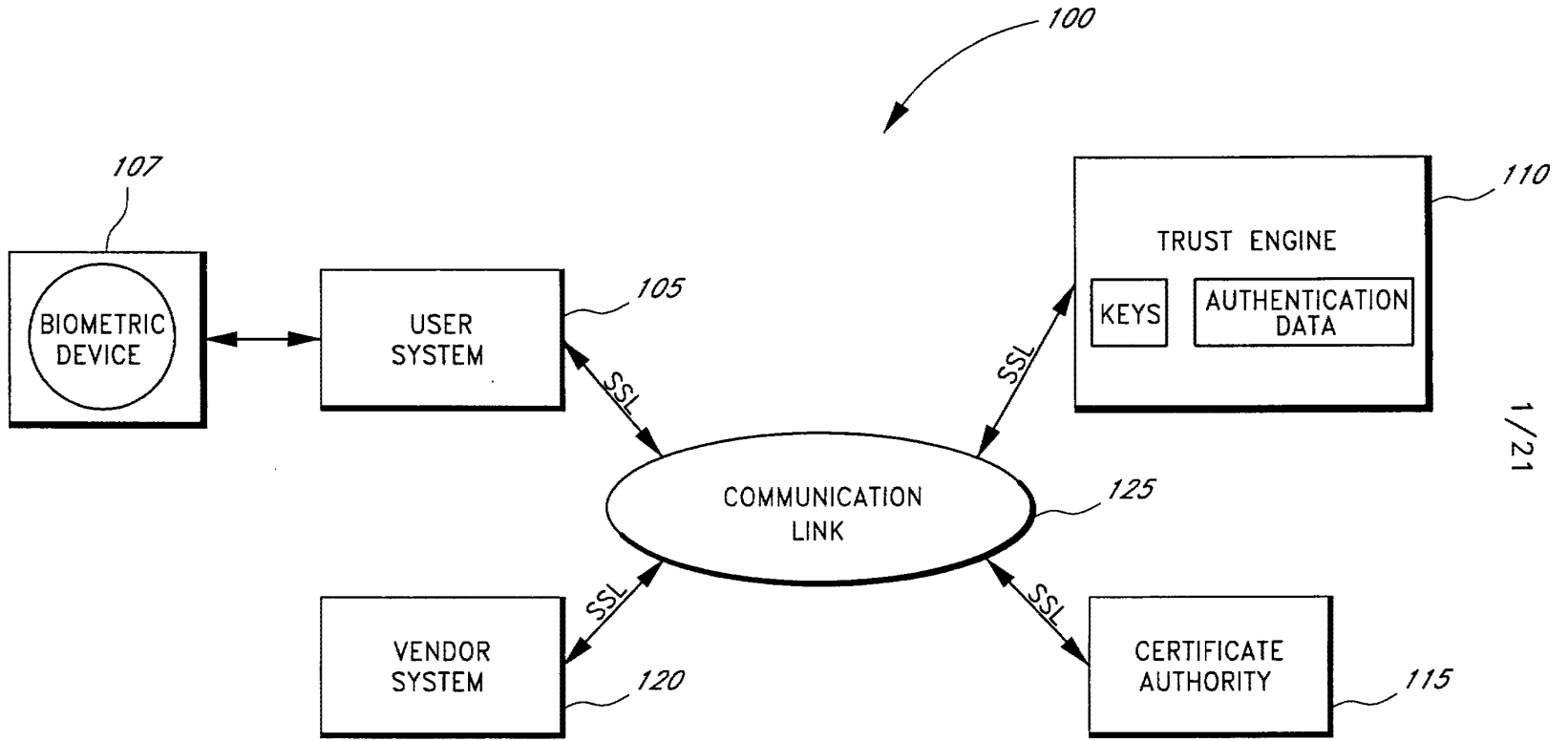
20       68.    An apparatus as in Claim 67 wherein the security functions comprise authenticating a user on the user system.

25       69.    An apparatus as in Claim 67 wherein the cryptographic functions comprise providing encryption of a message.

70.    A method of providing a cryptographic service with a network accessible server, comprising:  
receiving a request for a cryptographic service from an application programming interface;  
preparing a request for cryptographic functions, the cryptographic functions being necessary to  
30   provide the requested cryptographic service to the application programming interface;  
sending the request for cryptographic functions to a server across a communications link;  
receiving a result based upon the request for cryptographic functions from the server;  
preparing a response to the request for the cryptographic service based upon the result received  
from the server; and  
35   sending the response to the request for the cryptographic server available to the application programming interface.

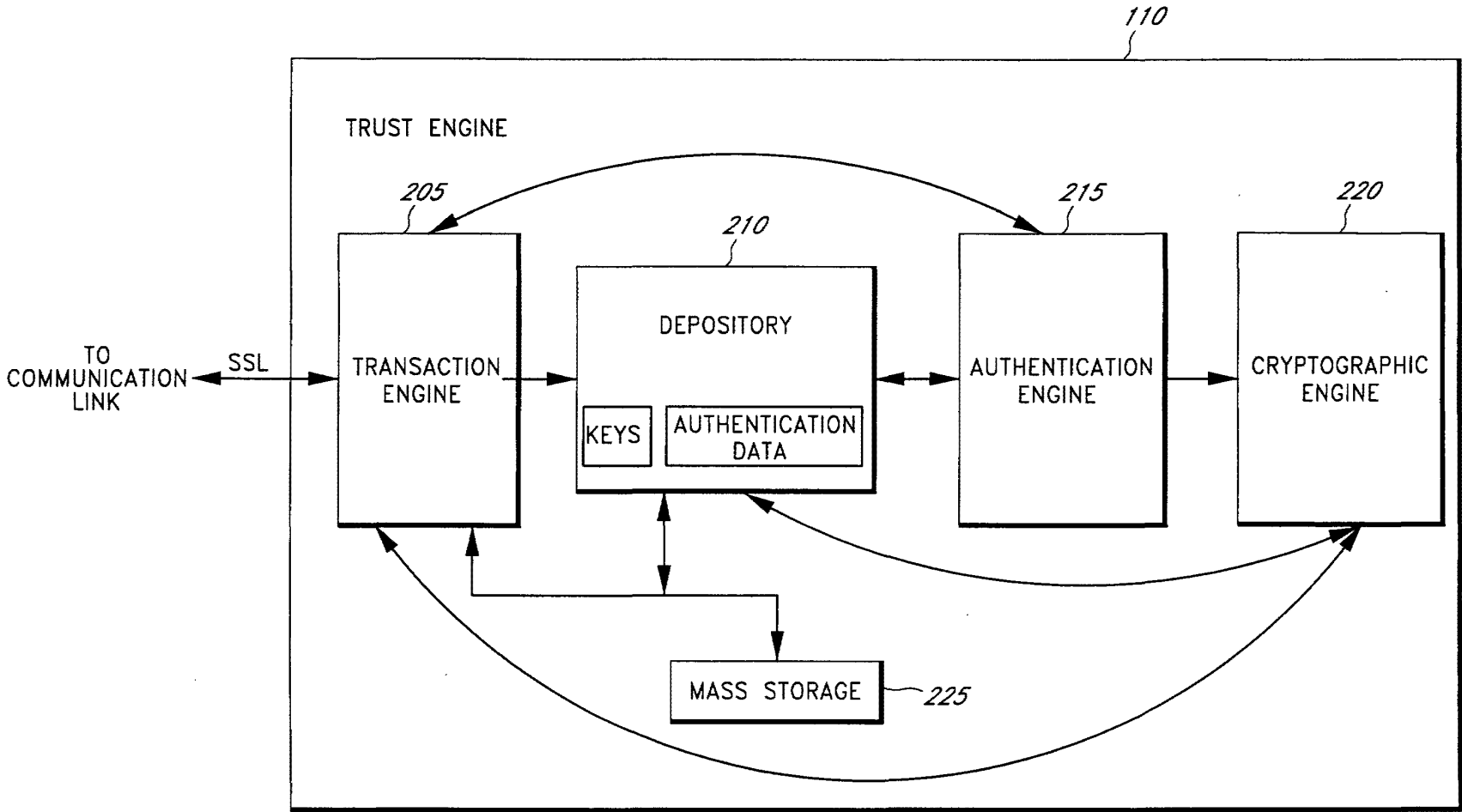
- 5
71. A method of performing cryptographic functions with a network-based cryptographic server, comprising:  
receiving an authentication request from an application programming interface; and  
processing the authentication request with a remotely located server.
72. A method as in Claim 71 further comprising sending authentication data to the remotely located server.
- 10
73. A method as in Claim 72 further comprising comparing the authentication data with enrollment data accessible by the remotely located server.
74. A method as in Claim 71 further comprising providing a response from the remotely located server based upon the authentication request.
- 15
75. A method of performing authentication functions with a network-based trust engine, comprising:  
receiving at a user computer, a request for an authentication function from an application programming interface; and  
routing the request to a network-based trust engine.
- 20
76. A method as in Claim 75 wherein the authentication function comprises providing digital certificates.
77. A method as in Claim 75 wherein the authentication function comprises verifying the authenticity of a signed document.
- 25
78. A method of interconnecting a cryptographic application programming interface with a network accessible resource, comprising:  
a cryptographic application programming interface on a client computer; and  
a cryptographic service provider module that is in communication with the cryptographic application programming interface wherein the cryptographic service provider is configured to perform cryptographic functions via a network accessible resource.
- 30
79. An apparatus for providing cryptographic functions to a user system from a remote trust engine comprising a cryptographic service provider module in communication with an application programming interface of the user system, the cryptographic service provider module receiving a request for cryptographic functions from the
- 35

application programming interface, and the cryptographic service provider module connected to a network accessible trust engine via a communication link, the trust engine providing a response to a requests for cryptographic functions from the cryptographic service provider module.

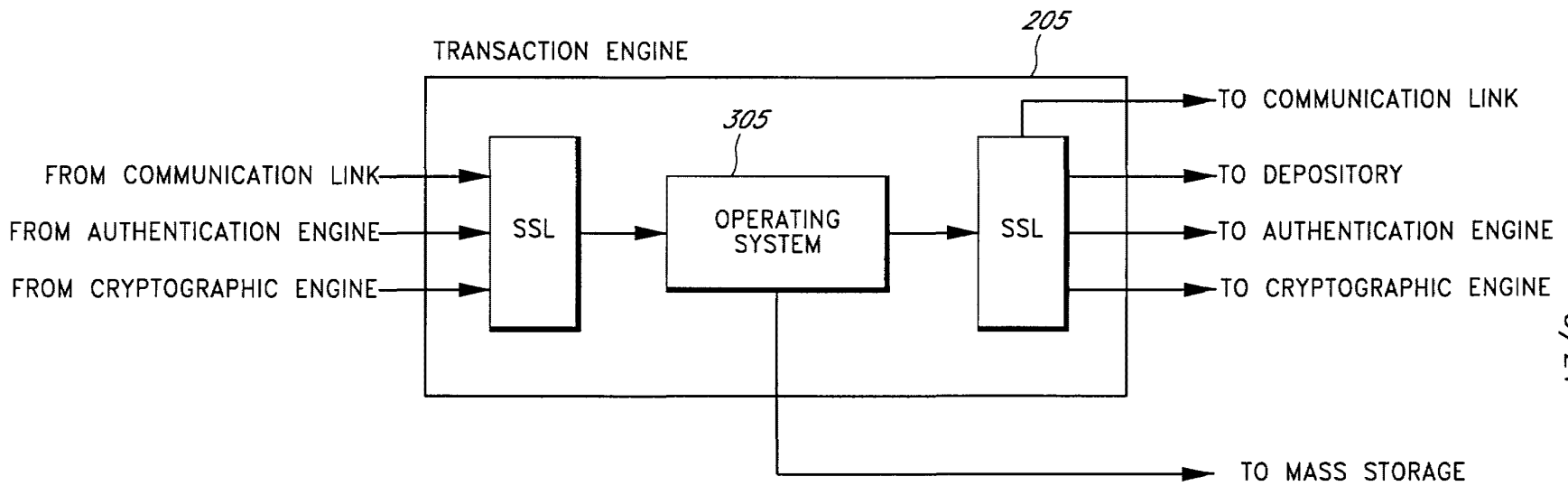


1/21

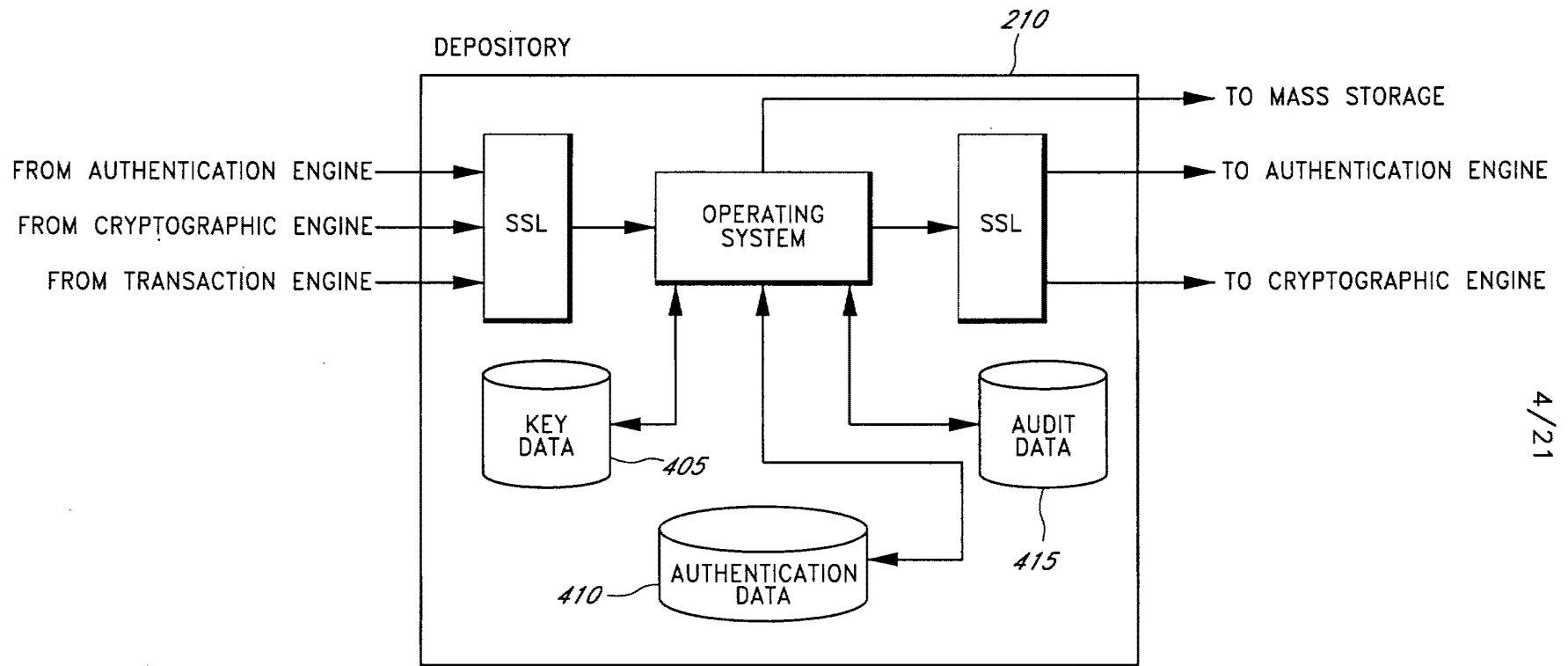
**FIG. 1**



**FIG. 2**



**FIG. 3**



**FIG. 4**

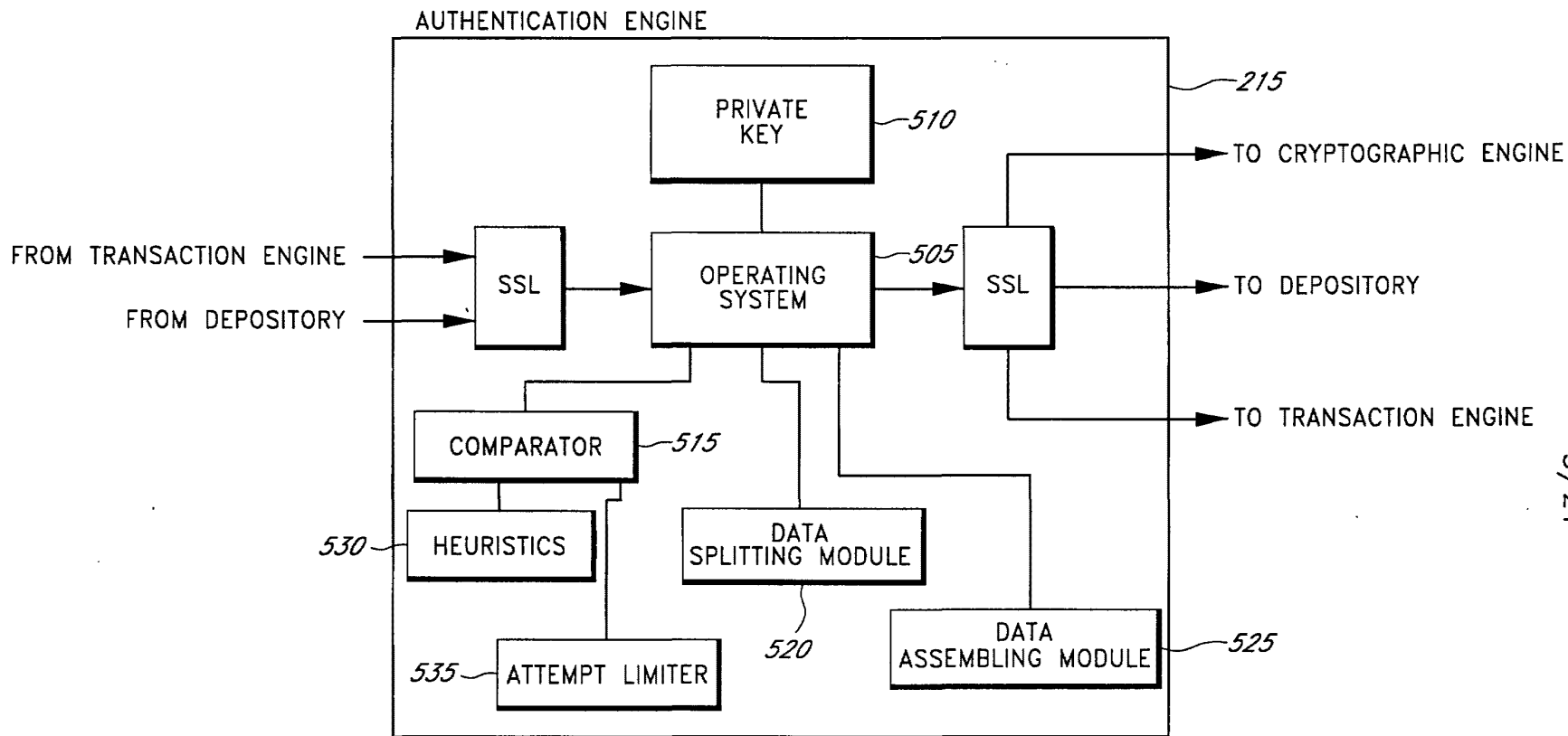


FIG. 5

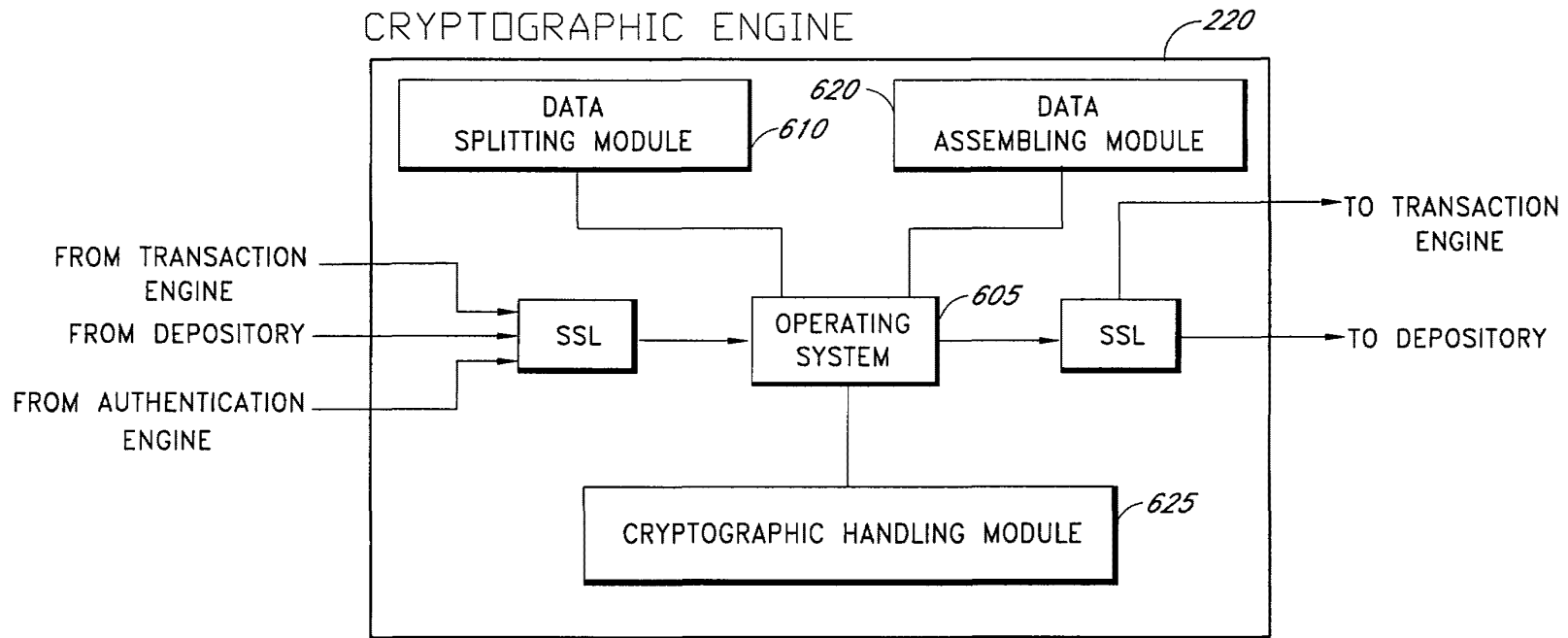
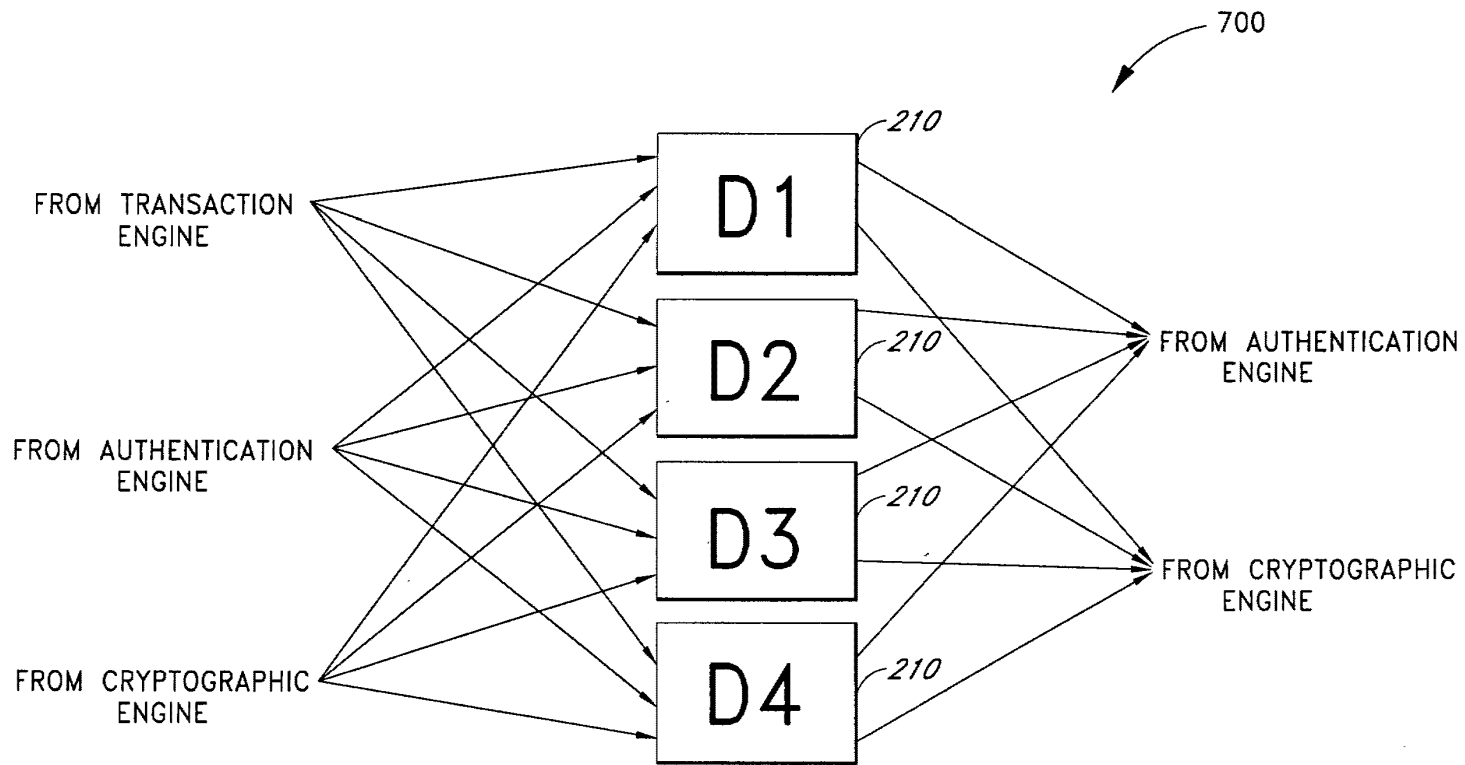
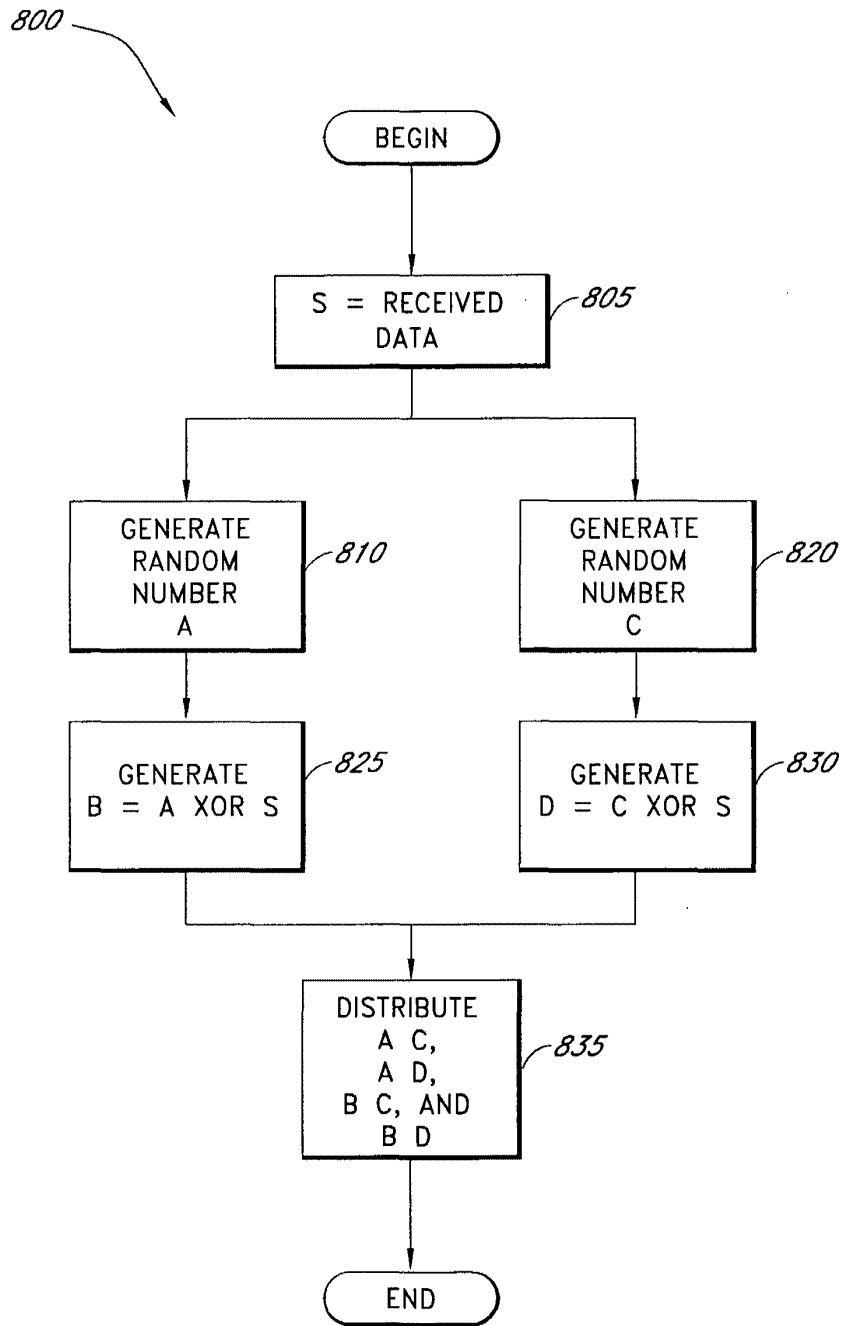


FIG. 6



**FIG. 7**

8/21



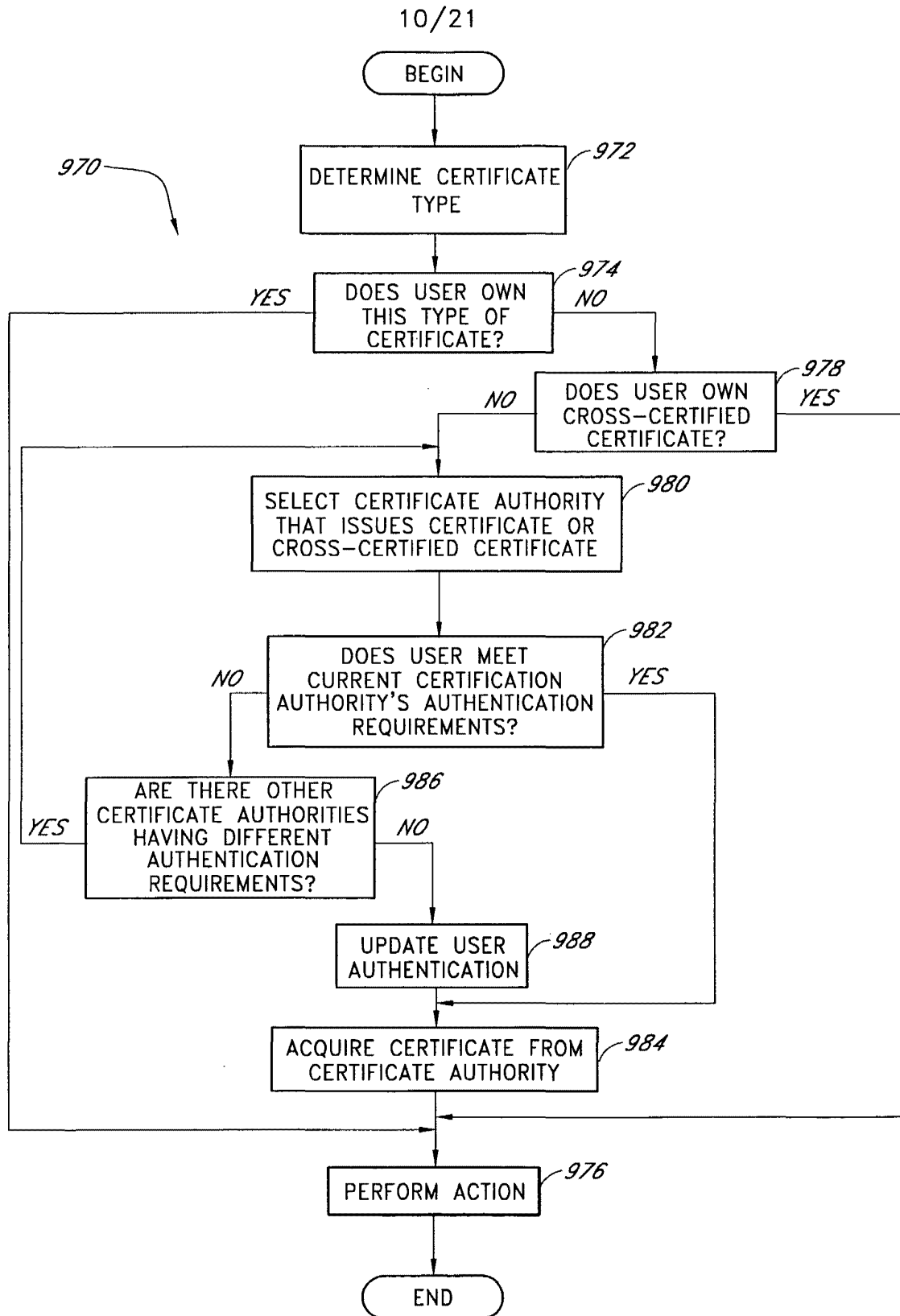
**FIG. 8**

9/21

900

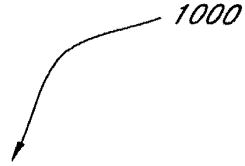
ENROLLMENT DATA FLOW			
SEND	RECEIVE	SSL	ACTION
905 USER	TRANSACTION ENGINE (TE)	1/2	TRANSMIT ENROLLMENT AUTHENTICATION DATA (B) AND THE USER ID (UID) ENCRYPTED WITH THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(UID,B))
915 TE	AE	FULL	FORWARD TRANSMISSION
920			AE DECRYPTS AND SPLITS FORWARDED DATA
925 AE	THE Xth DEPOSITORY (DX)	FULL	STORE RESPECTIVE PORTION OF DATA
WHEN DIGITAL CERTIFICATE REQUESTED			
930 AE	CRYPTOGRAPHIC ENGINE (CE)	FULL	REQUEST KEY GENERATION
935			CE GENERATES AND SPLITS KEY
945 CE	TE	FULL	TRANSMIT REQUEST FOR DIGITAL CERTIFICATE
950 TE	CERTIFICATION AUTHORITY (CA)	1/2	TRANSMIT REQUEST
955 CA	TE	1/2	TRANSMIT DIGITAL CERTIFICATE
960 TE	USER	1/2	TRANSMIT DIGITAL CERTIFICATE
TE	MS	FULL	STORE DIGITAL CERTIFICATE
965 TE	DX	FULL	STORE RESPECTIVE PORTION OF KEY

**FIG. 9A**



**FIG. 9B**

11/21



AUTHENTICATION DATA FLOW

	SEND	RECEIVE	SSL	ACTION
1005	USER	VENDOR	1/2	TRANSACTION OCCURS, SUCH AS SELECTING PURCHASE
1010	VENDOR	USER	1/2	TRANSMIT TRANSACTION ID (TID) AND AUTHENTICATION REQUEST (AR)
				AUTHENTICATION DATA (B') IS GATHERED FROM USER
1015	USER	TE	1/2	TRANSMIT TID AND B' WRAPPED IN THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE), AS (PUB_AE(TID, B'))
1020	TE	AE	FULL	FORWARD TRANSMISSION
				ENROLLMENT AUTHENTICATION DATA (B) IS REQUESTED AND GATHERED
1025	VENDOR	TRANSACTION ENGINE (TE)	FULL	TRANSMITS TID, AR
1030	TE	MASS STORAGE(MS)	FULL	CREATE RECORD IN DATABASE
1035	TE	THE Xth DEPOSITORY(DX)	FULL	UID, TID
1040	DX	AE	FULL	TRANSMIT THE TID AND THE PORTION OF THE AUTHENTICATION DATA STORED AT ENROLLMENT (BX) AS (PUB_AE(TID, BX))
1045				AE ASSEMBLES B AND COMPARES TO B'
1050	AE	TE	FULL	TID, THE FILLED IN AR
	TE	VENDOR	FULL	TID, YES/NO
1055	TE	USER	1/2	TID, CONFIRMATION MESSAGE

**FIG. 10**

12/21

1100

SIGNING DATA FLOW			
SEND	RECEIVE	SSL	ACTION
USER	VENDOR	1/2	TRANSACTION OCCURS, SUCH AS AGREEING ON A DEAL
VENDOR	USER	1/2	TRANSMIT TRANSACTION IDENTIFICATION NUMBER (TID), AUTHENTICATION REQUEST (AR), AND AGREEMENT OR MESSAGE (M)
			CURRENT AUTHENTICATION DATA (B') AND A HASH OF THE MESSAGE RECEIVED BY THE USER (h(M')) IS GATHERED FROM USER
USER	TE	1/2	TRANSMIT TID, B', AR, AND h(M') WRAPPED IN THE PUBLIC KEY OF THE AUTHENTICATION ENGINE (AE) AS (PUB_AE(TID, B', h(M')))
TE	AE	FULL	FORWARD TRANSMISSION
			GATHER ENROLLMENT AUTHENTICATION DATA
VENDOR	TRANSACTION ENGINE (TE)	FULL	TRANSMITS UID, TID, AR, AND A HASH OF THE MESSAGE (h(M)).
TE	MASS STORAGE (MS)	FULL	CREATE RECORD IN DATABASE
TE	THE Xth DEPOSITORY(DX)	FULL	UID, TID
DX	AE	FULL	TRANSMIT THE TID AND THE PORTION OF THE AUTHENTICATION DATA STORED AT ENROLLMENT (BX),AS (PUB_AE(TID, BX))
			THE ORIGINAL VENDOR MESSAGE IS TRANSMITTED TO THE AE
TE	AE	FULL	TRANSMIT h(M)
			AE ASSEMBLES B, COMPARES TO B' AND COMPARES h(M) TO h(M')
AE	CRYPTOGRAPHIC ENGINE (CE)	FULL	REQUEST FOR DIGITAL SIGNATURE AND A MESSAGE TO BE SIGNED, FOR EXAMPLE, THE HASHED MESSAGE
AE	DX	FULL	TID, SIGNING UID
DX	CE	FULL	TRANSMIT THE PORTION OF THE CRYPTOGRAPHIC KEY CORRESPONDING TO THE SIGNING PARTY
			CE ASSEMBLES KEY AND SIGNS
CE	AE	FULL	TRANSMIT THE DIGITAL SIGNATURE (S) OF SIGNING PARTY
AE	TE	FULL	TID, THE FILLED IN AR, h(M), AND S
TE	VENDOR	FULL	TID, A RECEIPT=(TID, YES/NO, AND S), AND THE DIGITAL SIGNATURE OF THE TRUST ENGINE, FOR EXAMPLE, A HASH OF THE RECEIPT ENCRYPTED WITH THE TRUST ENGINE'S PRIVATE KEY (Priv_TE(h(RECEIPT)))
TE	USER	1/2	TID, CONFIRMATION MESSAGE

1103  
1105  
1110  
1115  
1120  
1125  
1130  
1135  
1140

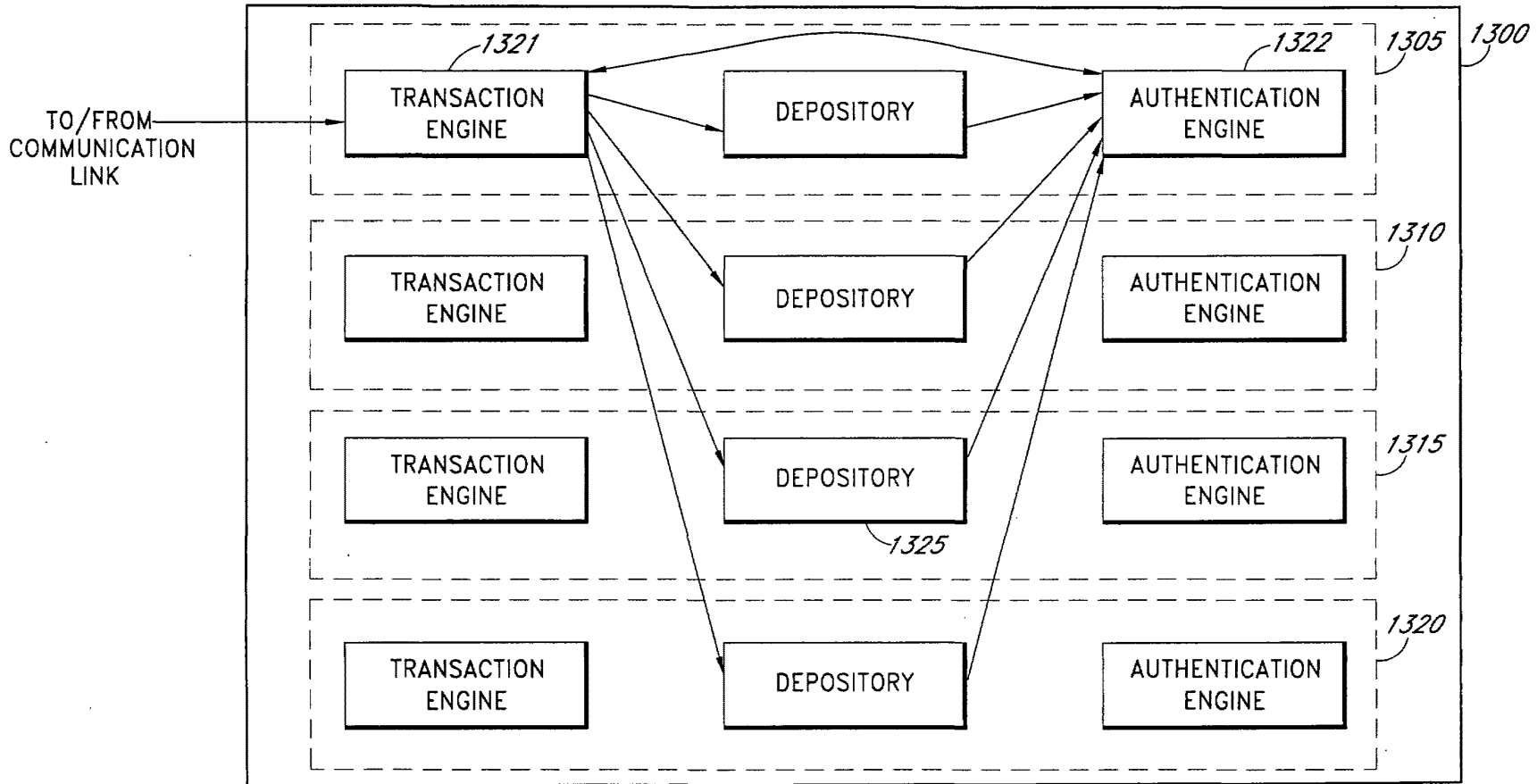
FIG. 11

1200

ENCRYPTION/DECRYPTION DATA FLOW				
SEND	RECEIVE	SSL	ACTION	
DECRYPTION				
			PERFORM AUTHENTICATION DATA PROCESS 1000, INCLUDE THE SESSION KEY (SYNC) IN THE AR, WHERE THE SYNC HAS BEEN ENCRYPTED WITH THE PUBLIC KEY OF THE USER AS PUB_USER(SNYC)	
			AUTHENTICATE THE USER	
1205 1210 1215	AE	CE	FULL	FORWARD PUB_USER(SYNC) TO CE
	AE	DX	FULL	UID, TID
	DX	CE	FULL	TRANSMIT THE TID AND THE PORTION OF THE PRIVATE KEY AS (PUB_AE(TID, KEY_USER))
1220				CE ASSEMBLES THE CRYPTOGRAPHIC KEY AND DECRYPTS THE SYNC
1225 1230	CE	AE	FULL	TID, THE FILLED IN AR INCLUDING DECRYPTED SYNC
	AE	TE	FULL	FORWARD TO TE
	TE	REQUESTING APP/VENDOR	1/2	TID, YES/NO, SYNC
ENCRYPTION				
1235 1240	REQUESTING APP/VENDOR	TE	1/2	REQUEST FOR PUBLIC KEY OF USER
1245	TE	MS	FULL	REQUEST DIGITAL CERTIFICATE
	MS	TE	FULL	TRANSMIT DIGITAL CERTIFICATE
1250	TE	REQUESTING APP/VENDOR	1/2	TRANSMIT DIGITAL CERTIFICATE

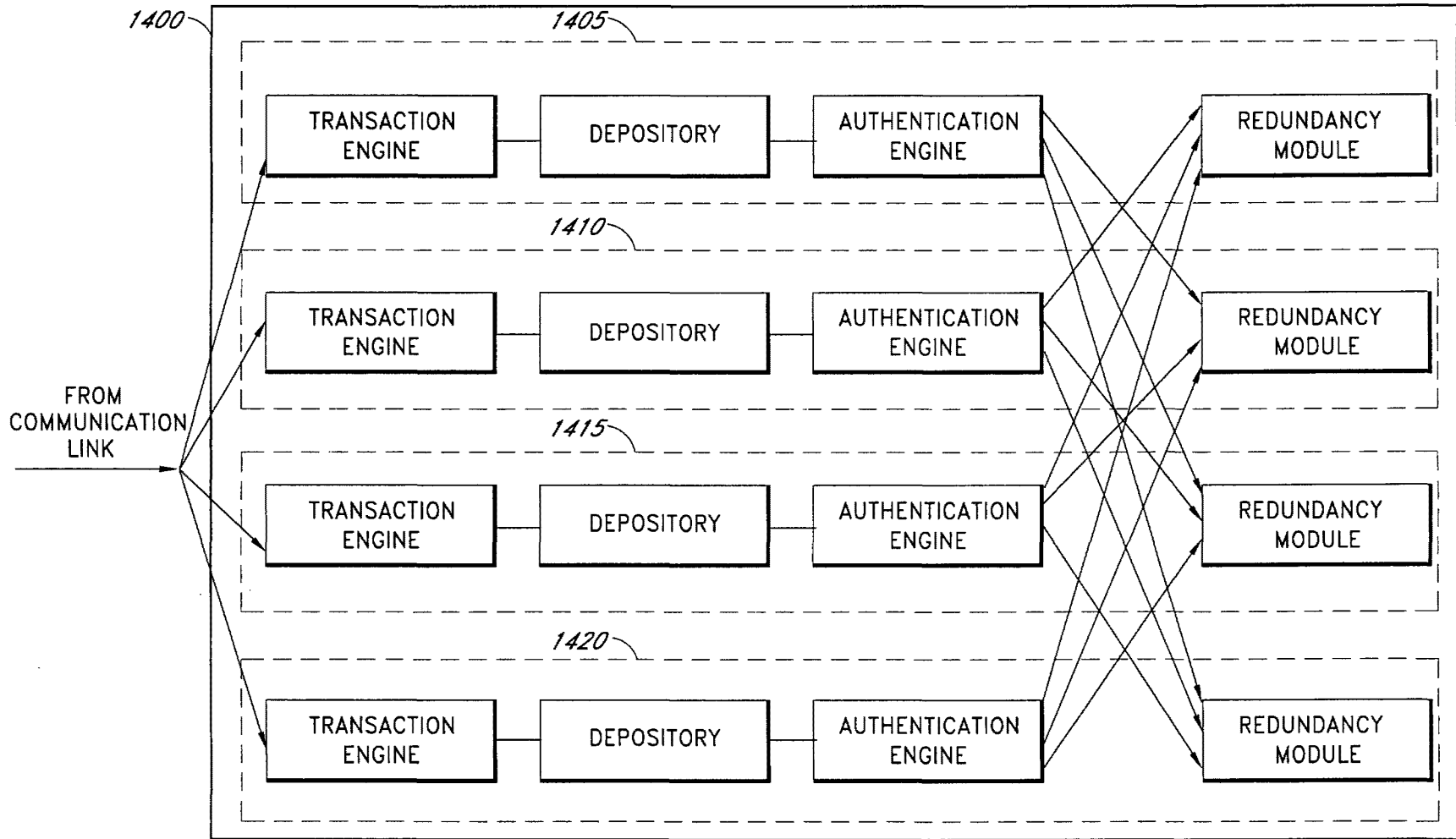
**FIG. 12**

14/21



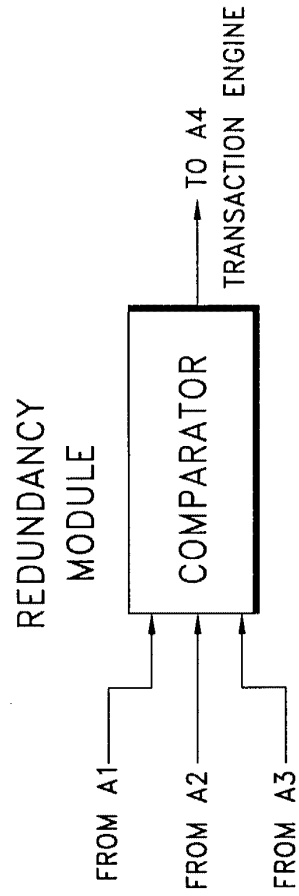
**FIG. 13**

15/21



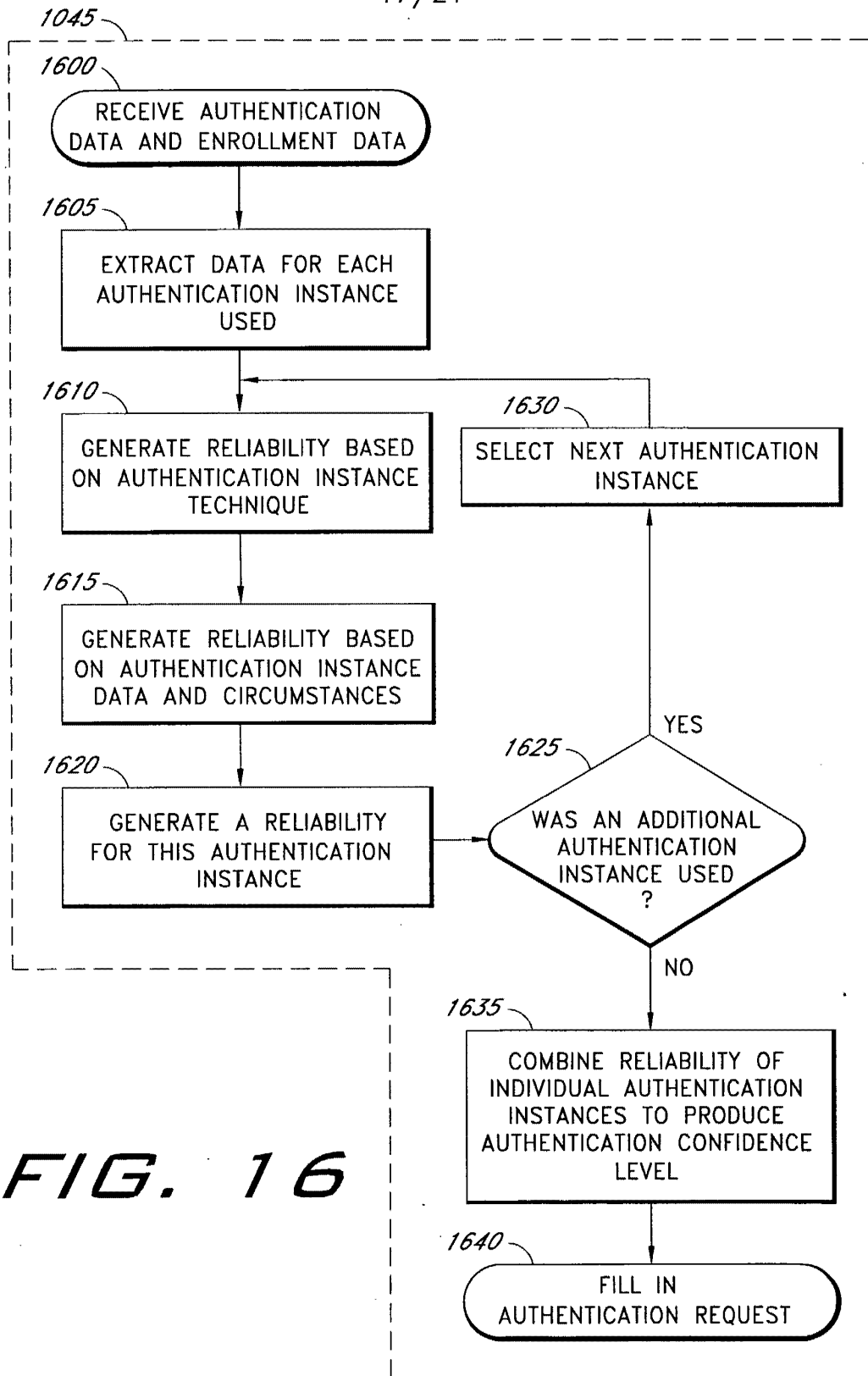
SUBSTITUTE SHEET (RULE 26)

FIG. 14

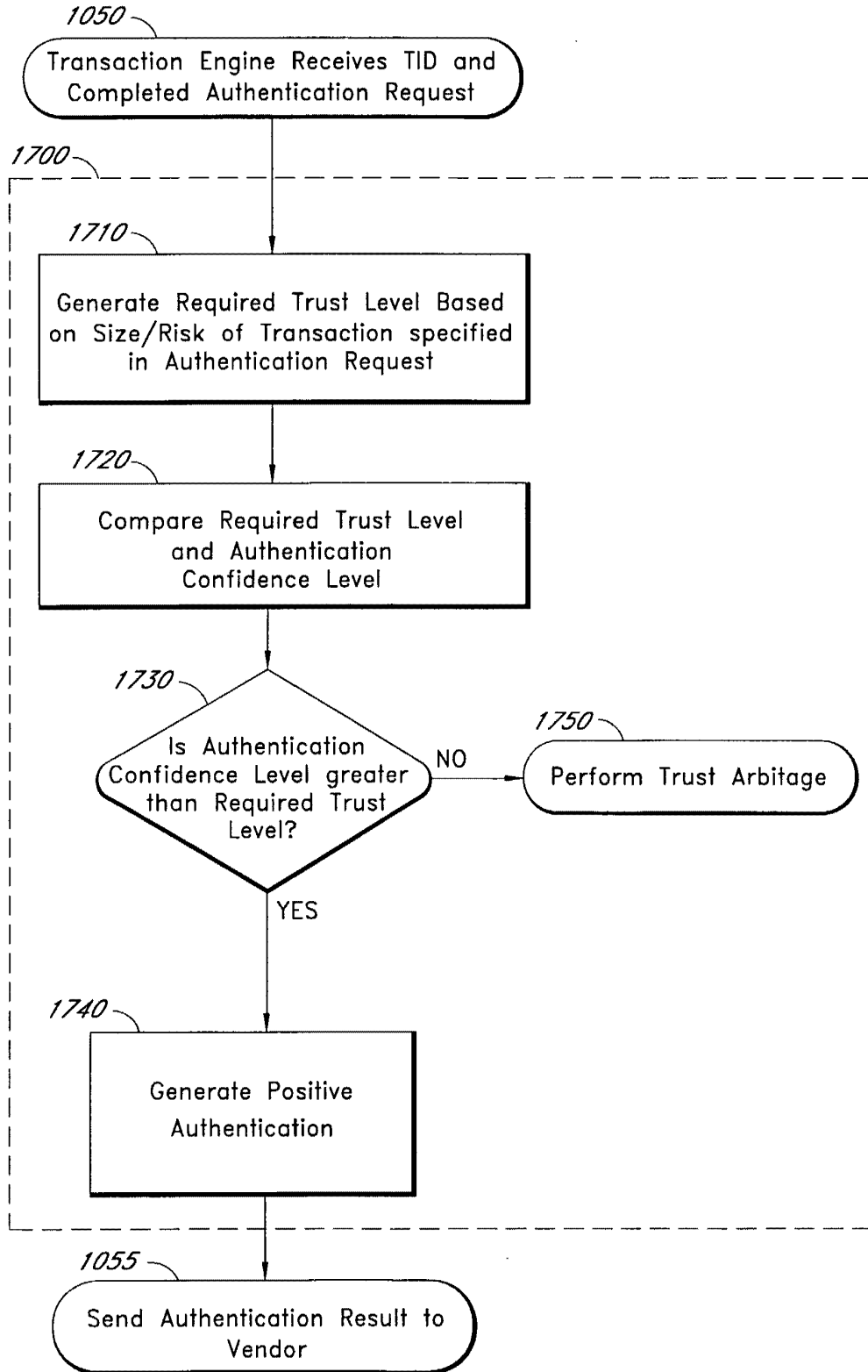


**FIG. 15**

17/21

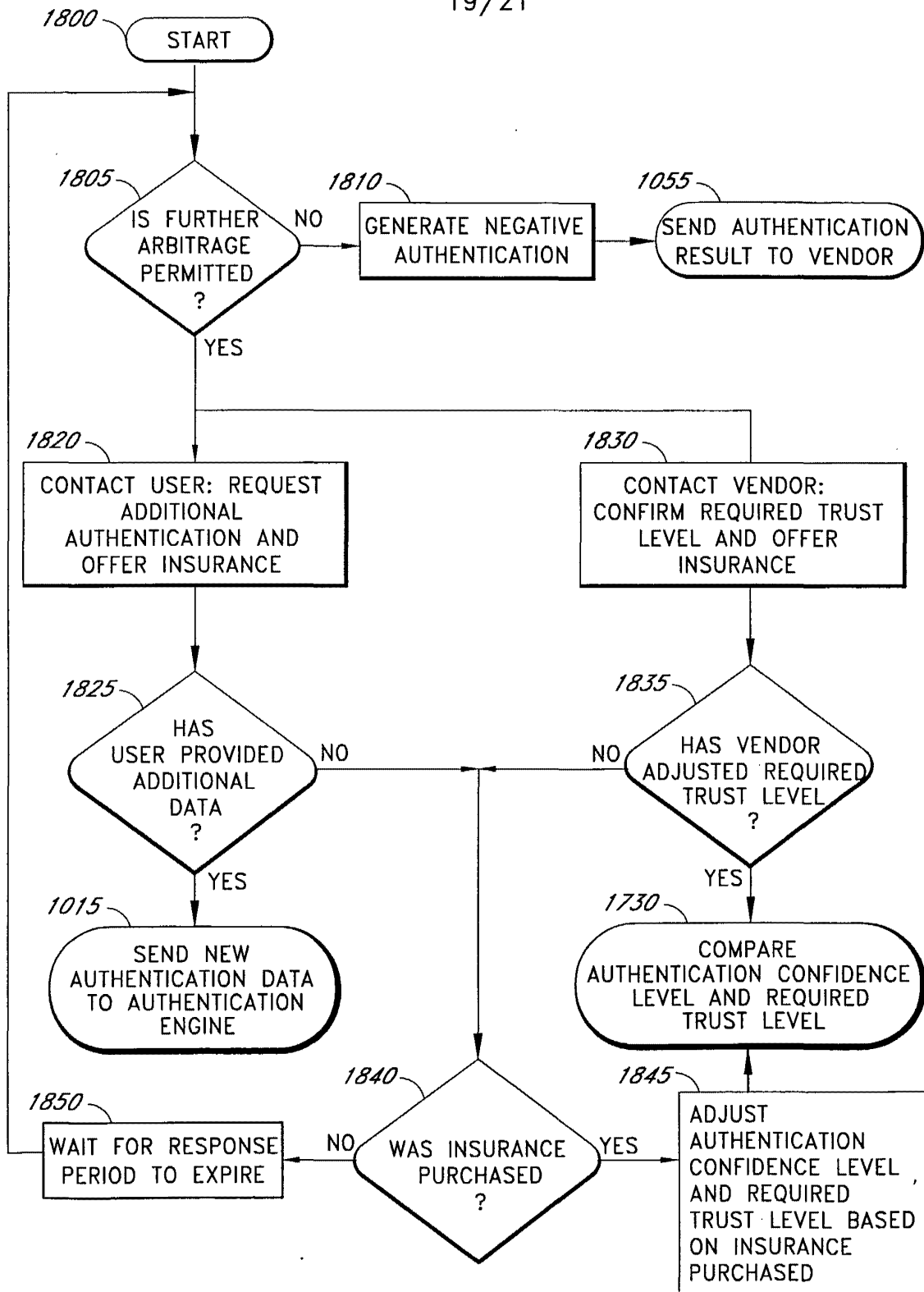


**FIG. 16**

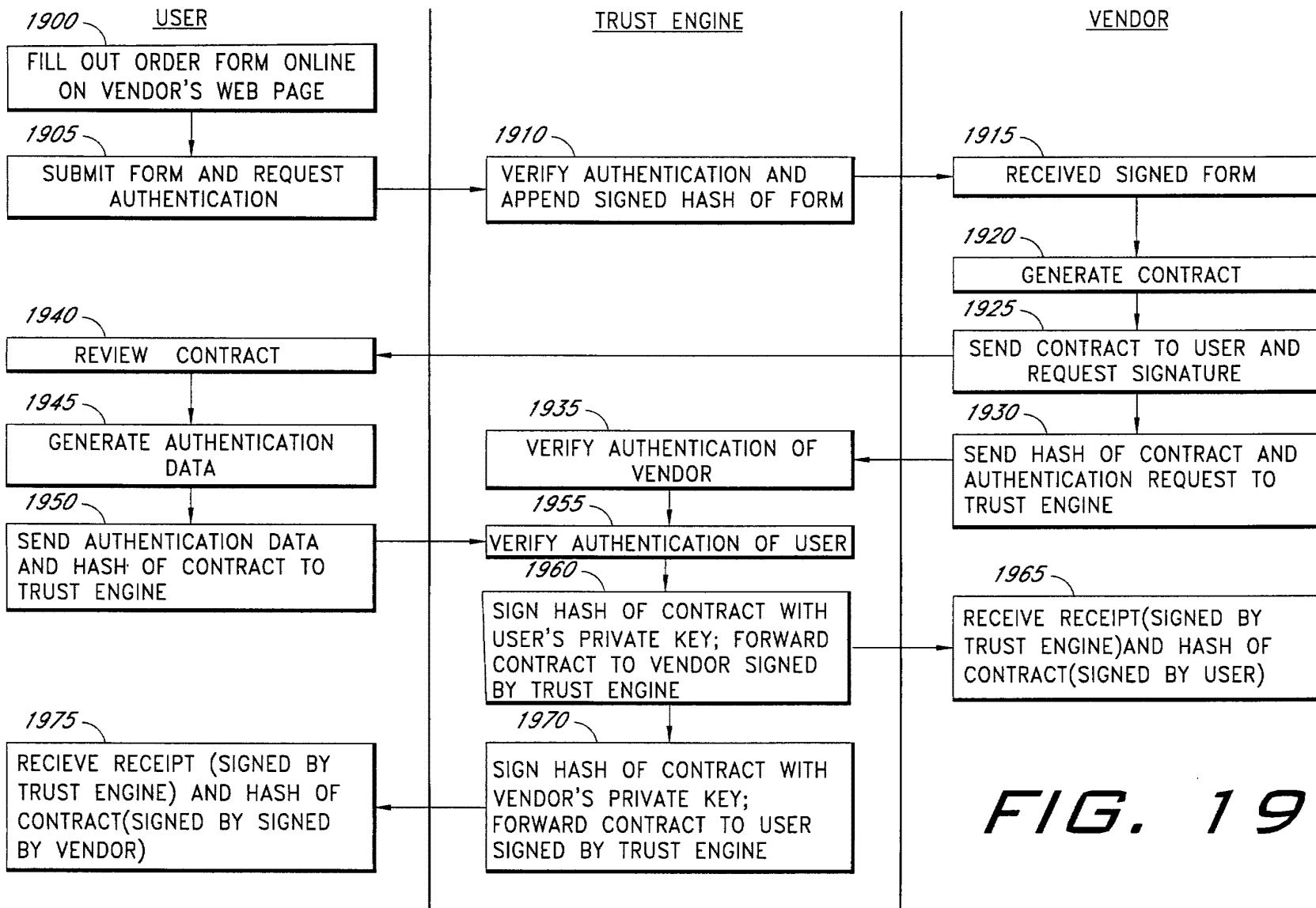


**FIG. 17**

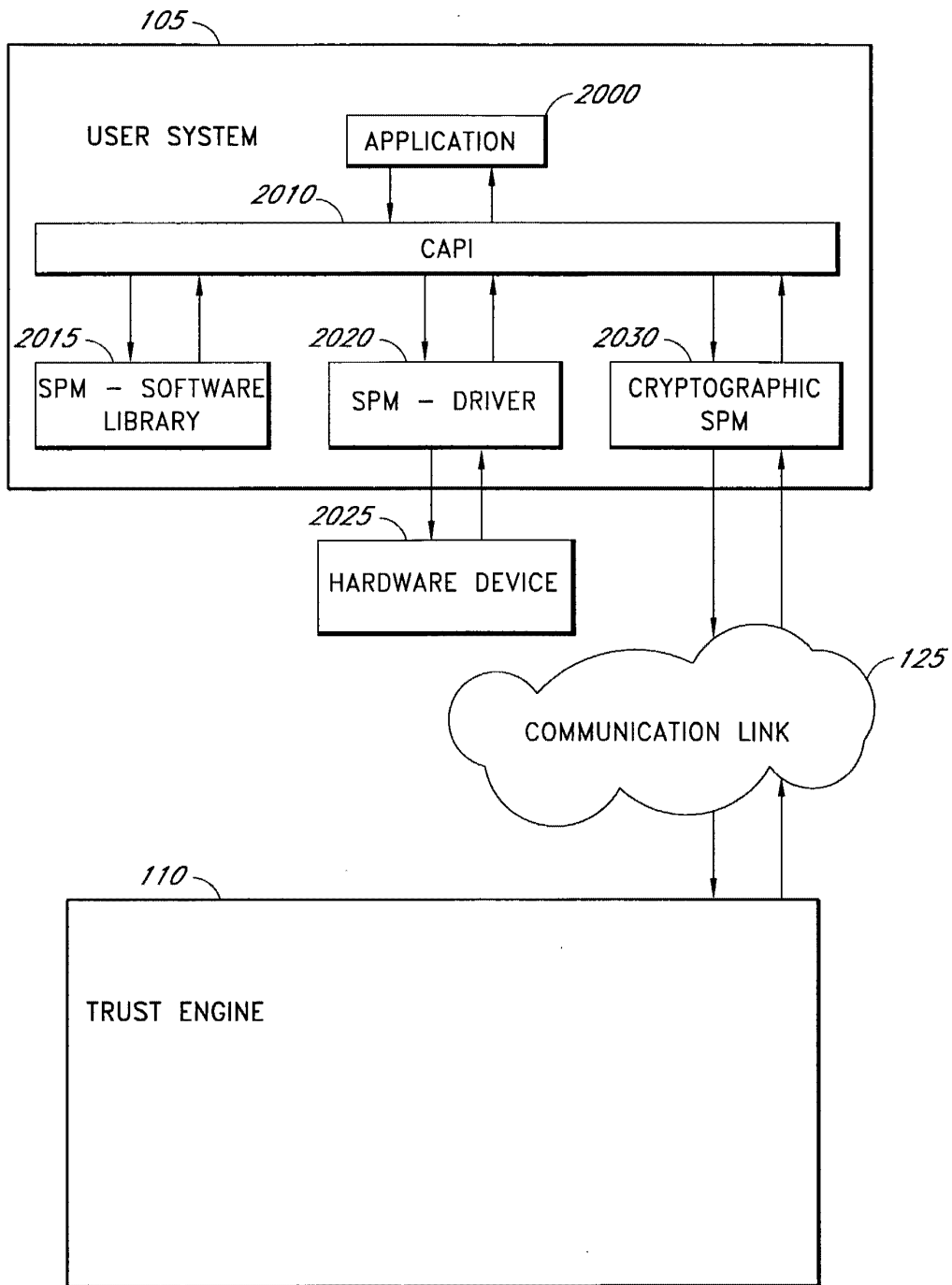
19/21



**FIG. 18**



**FIG. 19**



**FIG. 20**