

SYSTEMS AND METHODS FOR USING CIPHER OBJECTS TO PROTECT DATA

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application No. 61/569162, filed December 9, 2011, the entire disclosure of which is hereby incorporated by reference herein for all purposes.

BACKGROUND

Current techniques for protecting data have certain drawbacks. When information is outside of a trusted environment, like a firewall, it is typically protected by encryption. In current techniques, encryption keys must be present within an application or revealed to an application for encrypted data to be useful, thereby giving up protection and confidentiality. Encryption keys can be stolen via a brute force assault, or can be stolen via social engineering or other means. Further, once an encryption key is shared and the data unlocked, control of the data is lost. Even when data is within a trusted environment, such as behind a firewall and/or the like, it is vulnerable to attack or misuse, as files are available to anyone with access to their storage location. Protecting information traditionally requires teams of people with expertise in networks, telecommunication, and servers coordinating efforts on an enterprise scale to achieve a level of security which nevertheless can be breached by exploiting flaws in existing products.

Typical data encryption relies on algorithms which are run in a predetermined sequence to encrypt and then run in the reverse sequence to decrypt. There may also be a process of moving pieces of data in a static pattern to cloak it, and then reversing the process to reveal the complete, unencrypted file. With this method, an attacker who understands the encryption algorithm used to encrypt data can break the encryption by reversing the encryption process.

Fully homomorphic encryption attempts to remove the trust aspect of a relationship, making trust between parties an irrelevant factor. For example, one party can send their data to an outsourcer for storage or processing without trusting what the outsourcer might do with it, as the outsourcer is only given access to an encrypted version of the data to perform processing that does not require decryption. However, fully homomorphic encryption is too cumbersome to be practical.

Another traditional technique for protecting data is the use of dynamic controls. Dynamic controls are application dependent, such as protected PDF files generated and used by document viewing and editing software produced by Adobe®, and/or the like. Traditional dynamic controls are dependent on the application or reside within an application. Rules are executed by the application. One drawback to this method is that application-dependent rules may be overridden (as in the example of a protected PDF opened with Adobe® Acrobat®) or, a developer could write an application that ignores the rules imposed by the authoring application.

SUMMARY

This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to identify key features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

A computer-readable medium having an intelligent cipher transfer object stored thereon is provided. The intelligent cipher transfer object comprises a set of participants including a portable dynamic rule set. In response to execution by one or more processors of a computing device, the portable dynamic rule set causes the computing device to perform actions comprising receiving, from an agent, a request for access to a portion of a participant of the set of participants; verifying that the agent is authorized to access the requested participant portion; and providing access to the requested participant portion for the agent without providing access to other portions of the set of participants for the agent. A computer-implemented method of creating such an intelligent cipher transfer object and a computing device configured to execute the executable portions of such an intelligent cipher transfer object are also provided.

A computer-implemented method of protecting a set of participants is provided. A set of participants to be protected is obtained by a computing device. One or more cloaking patterns for protecting the set of participants are determined. A first cloaking pattern is used to protect a first subset of the set of participants, and a second cloaking pattern different from the first cloaking pattern is used to protect a second subset of the set of participants. The determined cloaking patterns are applied by the computing device to the set of participants to create a set of cloaked participants. The set of cloaked participants are added by the computing device to an intelligent cipher transfer object. A

computing device configured to perform this method and a computer-readable medium having computer-executable instructions stored thereon that, in response to execution by one or more processors of a computing device, cause the computing device to perform such a method are also provided.

5 A computer-implemented method of managing access to data protected by an intelligent cipher transfer object is provided. An access request from an agent to access a portion of a participant stored in the intelligent cipher transfer object is received by a computing device. A portable dynamic rule set within the intelligent cipher transfer object is activated by the computing device. At least one rule in the portable dynamic
10 rule set is executed by the computing device to evaluate the access request. In response to determining that the access request is permissible, access to the portion of the participant requested by the agent is provided without providing access to other participant portions. A computing device configured to perform this method and a computer-readable medium having computer-executable instructions stored thereon that,
15 in response to execution by one or more processors of a computing device, cause the computing device to perform such a method are also provided.

DESCRIPTION OF THE DRAWINGS

The foregoing aspects and many of the attendant advantages of embodiments of
20 the present disclosure will become more readily appreciated as the same become better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIGURE 1 is a schematic diagram that illustrates an exemplary embodiment of data protection according to various aspects of the present disclosure;

25 FIGURE 2 is a flowchart that illustrates an exemplary embodiment of a method of constructing an intelligent cipher transfer object (ICTO) according to various aspects of the present disclosure;

FIGURE 3 is a flowchart that illustrates an exemplary embodiment of a method of accessing data protected by an ICTO according to various aspects of the present
30 disclosure;

FIGURE 4 is a schematic diagram that illustrates an exemplary use case for an embodiment of the present disclosure;

FIGURE 5 is a schematic diagram that illustrates aspects of an exemplary workflow for an embodiment of the present disclosure; and

FIGURE 6 is a block diagram that illustrates an exemplary hardware architecture of a computing device suitable for use with embodiments of the present disclosure.

5

DETAILED DESCRIPTION

Embodiments of the present disclosure address these critical faults with previous data protection systems and methods. This technology fills a gap in existing protection schemes because existing schemes address perimeter defenses, user access (both users and their devices) and anomaly detection but are not attached to the data itself. If existing encryption is utilized on its own, flaws in key code management may reduce productivity or create yet other vulnerabilities by exposing keys that likewise need to be protected. Embodiments of the present disclosure provide a data-centric solution, meaning that the controls for data management, protection, and administration are grafted into, and become part of, each data set and directly oversee the data set's use. Though, in some embodiments of the present disclosure, some data is removed from protection to be usable by an authorized agent, the removal from protection is not predictable because it is not a reversal of the protection mechanism. The present disclosure presents an unpredictable and irreversible system and associated methods to retain dynamic, portable, independent, persistent, intelligent ownership of data over the life of the data's existence. This system is capable of protecting data while the data is stored or in transit, and in the hands of trusted data users or untrusted data users.

In some embodiments of the present disclosure, the data protection scheme is embedded within, grafted to, and maintained within the data-set. The data protection scheme may also create an audit trail of attempts to access the data. Known or authorized users of the data are noted in a log, while unknown parties or other unauthorized attempts to access the data are logged and can be displayed to the data owner in real time. If an unauthorized party attempts to access the data, the protection scheme can defend the data, take offensive action against the intrusion, alert the data owner to the unauthorized attempt, and/or take any other appropriate action.

The data owner sees the protection scheme as a simple and lightweight management tool that consistently validates the relationship of the parties to the data. From an attacker's point of view, the system is unpredictable because every authorized

party has its established identity incorporated into the protection scheme. A unique protection scheme may be provided for each combination of owner, user, and dataset; this means that the method by which data is revealed to Authorized Party A would not be the way data is revealed to Authorized Party B.

5 In some embodiments, different techniques may be used to protect the data and to access the protected data. For example, an irreversible protection scheme may be used to combine multiple pieces of data into a single digital mixture, while a selective retrieval scheme may be used to selectively retrieve pieces of data from the single digital mixture without obtaining access to other protected data in the digital mixture. In such an
10 embodiment, when the data is properly accessed, it is selectively revealed, based upon the owner's wishes for the authorized recipient; no steps are retraced to reveal information, and the original protection scheme used to combine the multiple pieces of data may not be reversible other than with respect to individually requested pieces of data. In other words, even though pieces of data stored by the protection scheme may be accessed, the
15 totality of the digital mixture is not accessible in such a way that allows the totality of the original contents to be reconstituted. Embodiments of the present disclosure are configured to positively identify any user, and the data owner controls which pieces of data to which the identified user can gain access. Unauthorized parties, whether inside or external to the intended recipient, can never access the data in its unprotected form.
20 Embodiments of the present disclosure unequivocally confirm the identity of a trusted party before providing access to ensure data security. Reversing or reverse engineering the protection scheme cannot yield the original results.

 In some embodiments of the present disclosure, rules are executed by an executable portion of the digital mixture, ensuring the absolute wishes of the data owner
25 are enforced without relying on a third party. The protection scheme is not dependent on an application or an operating system to protect the data. The protection scheme is independent of operating system, environment, and application. Methods in the protocol are implemented in executable code stored in the data mixture, and are executed in response to detecting a request by the user to access the data through a structured API.
30 Furthermore, the data can be of any type: text, audio, video; and, of any kind of container or environment: buffer, directory or file. Any attempt to access the data other than through the API will be foiled by the applied cloaking patterns, which will be undeterminable by any component outside of the components implementing the API.

When attempting to access the data through the API, the protection scheme ensures that users are only able to access data as permitted by the data owner.

FIGURE 1 is a schematic diagram that illustrates an exemplary embodiment of data protection according to various aspects of the present disclosure. A dynamic participant controller 110, or "mixer," identifies a set of digital ingredients ("participants") 101, including descriptions of authorized agents, device locations, rules for using the data, and/or other ingredients as discussed further below. By mixing these ingredients, the mixer 110 forms a cloaked entity, an intelligent cipher transfer object (ICTO) 115. The ICTO 115 may also be called a "digital mixture." As discussed herein, one of ordinary skill in the art will recognize that the terms "ICTO" and "digital mixture" may be used interchangeably. To a third party viewing the ICTO 115 directly, the ICTO 115 may simply appear to be a set of data. The ICTO 115 appears to the outside as a homogenous mixture without resembling or exposing the original ingredients. However, when accessed via an application implementing the API (such as the mixer 110, a client application (not illustrated), and/or the like), executable portions of the ICTO 115 are accessible to provide access to the data protected by the ICTO 115.

In some embodiments, the executable portions of the ICTO 115 may be stored at a determinable location within the ICTO 115 to allow an application implementing the API to easily find the executable portions. In some embodiments, additional protection may be applied to the ICTO 115 by storing one or more executable portions of the ICTO 115 at variable locations within the ICTO 115. While these variable locations make the executable portions of the ICTO 115 exceedingly difficult for an unauthorized user to find, a secure application implementing the API for accessing the ICTO 115 may be able to compute the variable locations for a given ICTO 115 based on a feature of the ICTO 115. For example, the secure application may read an attribute of the ICTO 115 such as a file size, a creation time, and/or the like, and may perform a calculation that determines the location using the attribute as a seed. By keeping the details of the calculation secret, the location of the executable portions of the ICTO 115 can likewise be kept secret.

The set of participants 101 may include object descriptions 102, mixture metadata 104, owner data 106, cloaking patterns 107, an identity module 109, and an intelligence module 111. In some embodiments, a combination of the identity module 109 and the intelligence module 111 may be considered together as a portable dynamic rule

set 108. The object descriptions 102 may include owner-supplied and owner-defined earmarks, data identifiers, and/or properties. Owner data 106 may include data that is to be protected within the ICTO 115, such as a document, a file, a buffer, a directory, a pointer to remotely stored data, and/or the like. In some embodiments, owner data 106 may be optional, if the ICTO 115 is merely used, for example, for a signature verification method that is not associated with underlying signed data. In some embodiments, multiple pieces of owner data 106 may be included within a single ICTO 115. In some embodiments, owner data 106 from multiple owners may be included within a single ICTO 115.

The cloaking patterns 107 specify various combinations of data protection and access techniques supported by the mixer 110. The data protection and access techniques included in cloaking patterns 107 may include techniques such as industry standard verified encryption, compression, randomization, normalization, and/or other techniques. Techniques suitable for use as cloaking patterns 107 are not limited to currently known techniques, but could include any privately or publicly available encoding and/or decoding technique, known now or developed in the future. Use of a cloaking pattern 107 to protect and/or access data may involve applying the combination of data protection and/or access techniques specified in the cloaking pattern 107 to the data.

The mixture metadata 104 provides organizational information for the digital mixture 115, such as virtual file system data containing directories, key codes, user files, signatures, and/or the like.

The identity module 109 may include dynamic identity attributes that uniquely identify protected agents in a transaction. In some embodiments, the identity module 109 may include data that represents a configuration of a computing device that may be given certain rights with respect to a protected object. The identity module 109 may contain specific information about hardware or software configurations installed on the computing device usable to identify the computing device. The identity module 109 may contain data including, but not limited to, CPU information including model numbers, number of cores, speed, and/or the like; a chassis serial number; manufacturer data; a volatile memory size; a non-volatile memory size; one or more storage device serial numbers and/or model numbers; installed software titles and/or version numbers, and/or the like.

In some embodiments, a transaction is an atomic action using the ICTO 115 in which one or more agents securely exchange data within a given context and with a specified intent. Authorized agents may include human and non-human entities, such as a human user, a unique mechanical object, a unique electronic object, a unique software or program object, and/or the like. Dynamic identity attributes contained in the ICTO 115 may be modified by the intelligence module 111 within or during the course of an interaction with the ICTO 115, and may include application-specified identifiers, account identifiers, biometric signatures, device and/or location signatures, temporal data, cryptographic keys, and/or the like. In some embodiments, a location signature may include data from a geolocation technology, such as GPS, GSM network locating, IP address locating, dead reckoning, and/or the like. The location signature may include a longitude, latitude, an altitude, an approximate street address, and/or the like. Additional location data such as street, city, state, country, postal code, and/or the like may also be present. In some embodiments, the temporal data may include a timestamp and/or the like, which may allow rules or other intelligent code to enforce timers, expirations, dynamic keys, and/or the like. The temporal data may include a simple date/time value, or may include a complex schedule comprising timestamp ranges and/or other scheduling guidelines.

In some embodiments, each ICTO 115 includes at least one digital signature key. The digital signature key may be validated using an external digital certificate available to the mixer 110. During access of the ICTO 115, the mixer 110 validates the digital signature key using the external digital certificate and verifies that the digital signature key is valid for an agent currently accessing the ICTO 115. In some embodiments, multiple agents may sign off on the ICTO 115. In such an embodiment, the ICTO 115 may include a chain of signature keys, wherein each signature key may be associated with a separate external digital certificate for validation. For example, an ICTO 115 may be used by an owner to create a protected file for a transfer to multiple agents wherein each agent may access different sections of the file but not the entire file, either simultaneously or sequentially. Both the owner and the agents may have to provide valid digital signatures to allow the transaction to proceed.

The intelligence module 111 may include dynamic rule sets capable of recording and communicating access data and other relevant history; along with intelligent code that provides configurable functionality for performing actions to protect the ICTO 115.

Rules may be provided at object creation time. However, in some embodiments, a rule may have a capability to modify itself or other rules for a previously created ICTO 115. In some embodiments, a rule may have a capability to create additional rules. For example, a rule may determine, from identity data, that additional protection is desirable for a given ICTO 115. The rule may then create additional encryption and/or decryption rules to be applied. The rules are protected and contained within the ICTO 115. In some embodiments, the rules may only be executable by an executable portion of the intelligence module 111, and/or may be written in a proprietary language and stored in compiled or binary form. Based on the rules and requirements of the identity module 109, the intelligence module 111 acts on its rules and requirements. Application-specified identifiers may vary from access to access, and may vary depending on a type of agent. For example, for a human user, application-specified identifiers may include account keys, transaction information, context keys, associated intents, and/or the like. For an electronic object, a digital asset, or any other potential agent, application-specified identifiers may also include an IP address, a URL, a file specification, and/or the like.

In some embodiments, rules have read/write access to the participants 101, even while the participants 101 are protected by the ICTO 115. In other words, a rule may read and write data to the mixture metadata 104 and the owner data 106 of the ICTO 115. This may be useful for recording access information such as date, time, place, and the like, and/or to destroy the data if an attack is detected. Some examples of decisions made or actions taken by intelligent code within rules may include, but are not limited to: evaluating object content and context for validity; challenging a participant for proof of identity; interacting with client code; contacting a server for validation; causing the ICTO 115 to self-destruct; maintaining a history of object access and sending the history information to a server; allowing on-line and/or off-line object access; creating new rules based on dynamic server updates; encrypting and decrypting data; mangling and unmangling data; and/or the like.

The use of portable dynamic rules may have various benefits. For example, pre-encryption and pre-decryption rules may provide dynamic salt and encryption keys based on participant-specified criteria. Such dynamic keys may be based on temporal data, environment data, or any other algorithm specified in a pre-encryption rule. As another example, rules may access encrypted identity artifacts within the ICTO 115 in order to validate the client without exposing the unprotected data to the world. As yet another

example, because the rules are portable and are therefore included within the ICTO 115, rules may be written in such a way as to allow the ICTO 115 to be fully protected from unauthorized access even when offline. As a further example, rules may add nested protection. If the ICTO 115 protects a document that is meant to be read by a single client within one hour of creation, a rule may implement the timer and issue a self-destruct mechanism.

As stated above, the mixer 110 uses a portable dynamic rule set 108 to form a mixture of the object descriptions 102, the mixture metadata 104, the owner data 106, the cloaking patterns 107, the identity module 109, and the intelligence module 111 that makes up an ICTO 115. In some embodiments, various components of the ICTO 115 may be marked by encoded checksums to detect tampering. For example, the entire ICTO 115, the rules, the owner data, and/or the user data may each be validated by a checksum. The checksum may be a hash value generated based on the contents of the checksum target. In some embodiments, the algorithm used to generate the checksum is sensitive enough to reliably detect a change of a single bit value of even a large document. Some suitable algorithms include MD5 and SHA, though any other suitable algorithm may be used. Each checksum may be appended, prepended, or otherwise combined with the checksum target for storage, or may be stored in a separate location.

FIGURE 2 is a flowchart that illustrates an exemplary embodiment of a method 200 of constructing an ICTO 115 according to various aspects of the present disclosure. While the illustrated method 200 describes creation of a relatively simple ICTO 115, one of ordinary skill in the art will understand that similar techniques may be used to create much more complex ICTOs 115. In some embodiments, the mixer 110 is configured to perform the method 200. In some embodiments, the method 200 is performed by a computing device, as described below, that is configured to provide the functionality of the mixer 110. One of ordinary skill in the art will recognize that the construction and utilization of the ICTO 115 is neither dependent on the type of said computing device nor on any operating system associated with said computing device, but may instead be constructed and utilized via any suitable means.

From a start block, the method 200 proceeds to block 202, where a set of common digital ingredients ("participants") is obtained. The common participants are participants 101 which may be used in more than one ICTO 115, or may at least have similar corresponding components in more than one ICTO 115, and are specified and/or

generated by the mixer 110 for inclusion in the ICTO 115. For example, the object descriptions 102, the mixture metadata 104, the cloaking patterns 107, the identity module 109, and the intelligence module 111 may all be common participants.

Next, at block 204, a dynamic participant controller ("mixer") 110 is initialized. In some embodiments, initializing the mixer 110 may include verifying that the mixer 110 is being executed by an expected or otherwise trusted application. At block 206, the mixer 110 receives one or more pieces of owner data 106 to be protected. As discussed above, in some embodiments the owner data 106 may be optional, and the access protection features of the ICTO 115 may be used to verify user identities and/or obtain signatures from users.

The method 200 proceeds to block 208, where the mixer 110 causes a portable dynamic rule set 108 to be executed. At block 210, an intelligence module 111 of the portable dynamic rule set 108 determines one or more identity-based cloaking patterns to be used to protect participants 101, and at block 212, the mixer 110 applies the one or more cloaking patterns to the participants 101, creating a set of cloaked participants.

The portable dynamic rule set 108 determines a cloaking pattern to be applied to each participant 101 based on the desires of the owner of the data to be protected. Different cloaking patterns may be applied to each participant 101. Further, each participant 101 may be protected using separate cloaking patterns for access by different agents. In other words, a participant 101 such as owner data 106 may be protected by a first cloaking pattern for access by a first agent, and protected by a second cloaking pattern for access by a second agent. The selection of cloaking patterns may be based on an attribute of the participant 101 to be protected, an attribute of the agent to be given access to the data, a location, an intent, and/or any other suitable piece of information. Selection of a cloaking pattern may include selecting from a pre-existing cloaking pattern, and/or may include creating a new cloaking pattern from a combination of protection techniques supported by the mixer 110. Records of the applied cloaking patterns may be stored in the mixture metadata 104.

Cloaking patterns describe transformations applied to a participant 101 to protect the participant 101 within the ICTO 115, and how those transformations may be reversed to access the participant 101. The transformations may include, but are not limited to, data compression, data normalization, and encryption/decryption. A given cloaking pattern may include one or more of these techniques, or other techniques not listed here.

Data compression may reduce the overall size of the ICTO 115, which may in turn improve transport times and bandwidth usage. Data compression may be performed by any suitable lossless compression algorithm including, but not limited to, DEFLATE, LZW, LZR, LZX, JBIG, DjVu, and/or the like. Data normalization is performed by any
5 suitable process that places the data in a form that may efficiently be processed. In some embodiments, the data may be passed through a Base64 encoding algorithm to convert the data, whether binary or text format, into a normalized alphanumeric string. This is an example only, and should not be seen as limiting. In other embodiments, other algorithms may be used to normalize the data.

10 Cloaking patterns may also include one or more encryption/decryption techniques. The cloaking patterns may specify methods of deriving encryption keys, may specify particular encryption algorithms or key lengths, and/or may specify other configurable options such as time seeds, Xor encoding, and/or other industry standard encoding and decoding techniques for generating the cloaking scheme. In some embodiments,
15 encryption techniques may perform calculations other than encryption based on referenced content, such as deriving a hash value for the referenced content and/or the like. In some embodiments, the cloaking pattern may store a record of an encryption key or decryption key used, either in the cloaking pattern itself or elsewhere within the ICTO 115. When the cloaking pattern is used to access the protected information, the
20 encryption/decryption key is provided to the rule within the ICTO 115 to provide access to the information, but is not made available to the requesting user. In other words, the encryption keys and decryption keys are not made available to the users, and so their secrecy is maintained.

In some embodiments, a cloaking pattern may cause the identity module 109 and
25 the intelligence module 111 to apply separate encryption techniques to different components of the participants 101. For example, a first encryption rule, when executed, may identify and encrypt a first portion of the encrypted digital mixture 115 while leaving a second portion of the encrypted digital mixture 115 unchanged. A second encryption rule, when executed, may then identify and encrypt the second portion of the encrypted
30 digital mixture 115 using a different encryption algorithm, a different encryption key, and/or the like.

In some embodiments, the cloaking patterns and/or the portable dynamic rule set 108 may establish two or more nested layers of encryption. For example, execution of a

first encryption rule may encrypt a first portion of the encrypted digital mixture 115. Execution of a second encryption rule may then cause the encrypted first portion of the encrypted digital mixture 115 to be encrypted again, along with the first encryption rule and a corresponding first decryption rule. Hence, to later access the first portion of the encrypted digital mixture 115, a second decryption rule corresponding to the second encryption rule is executed to decrypt the doubly encrypted first portion of the encrypted digital mixture 115 and to obtain the first decryption rule. The first decryption rule is then executed to decrypt the first portion of the encrypted digital mixture 115 to generate a plaintext version of the first portion of the digital mixture 115.

Once the cloaking patterns have been applied to the participants 101 to create the set of cloaked participants, the method 200 proceeds to block 214, where the mixer 110 creates a digital mixture (ICTO) 115 and adds the set of cloaked participants to the digital mixture 115. In some embodiments, additional protection may be applied to the digital mixture 115 as a whole, such as shuffling of the data, additional encryption or digital signatures, and/or the like. The method 200 then proceeds to an end block and terminates.

One of ordinary skill in the art will understand that certain steps have been omitted from FIGURE 2 for ease of discussion. However, other steps not explicitly illustrated in FIGURE 2 may also be included in the method 200 without departing from the scope of the present disclosure. For example, if any errors are detected while applying the cloaking patterns or executing rules, the method 200 may stop, and may not produce a completed ICTO 115. As another example, in some embodiments, the owner data 106 may include one or more ICTOs as a way of providing nested protection. In some embodiments, rules within a nested ICTO may be provided with access to participant data 101 within the outer ICTO 115. In some embodiments, a rule within a first ICTO may cause a second ICTO to be created, and cause the first ICTO to be added to the second ICTO such that the first ICTO is nested inside of the second ICTO. Likewise, in some embodiments, a rule within a first ICTO may cause a second ICTO to be created, and cause the second ICTO to be added to the first ICTO such that the second ICTO is nested inside of the first ICTO.

FIGURE 3 is a flowchart that illustrates an exemplary embodiment of a method 300 of accessing data protected by an ICTO 115 according to various aspects of the present disclosure. After the ICTO 115 is activated, the ICTO 115 begins verification

and validation of its current environment, access attempts, authorized agents, and other conditions as specified in the rule set included in the portable dynamic rule set 108. This verification and validation may be performed once upon startup, continuously during an active period, periodically during an active period, or at any other suitable interval or in response to any suitable change in state. When rules and agent identity have been positively confirmed, the ICTO 115 permits access to authorized portions of itself while maintaining the homogenous essence of the mixture and protection of the rest of the data.

As with the method 200 described above, in some embodiments the mixer 110 is configured to perform the method 300. In some embodiments, the method 300 is performed by a computing device if one or more processors of the computing device execute computer executable instructions that cause the computing device to do so. As understood by one of ordinary skill in the art, the construction and utilization of the ICTO 115 is neither dependent on the type of said computing devices nor on any operating systems associated with said computing devices. The data protection protocol is embedded in the data set. An activated ICTO 115 can communicate with the data owner (information such as access attempts, alerts to unauthorized locations or unauthorized agents, notification of self-destruct or self-recreation) over the life of the data. Further, because the rules in the ICTO 115 may update themselves and other portions of the ICTO 115, the ICTO 115 may learn from its environment, and may change its future behavior based on that learning. The protection protocol can be customized and is unique to each owner, data set, and user combination, as specified in cloaking patterns.

From a start block, the method 300 proceeds to block 302, where a portable dynamic rule set 108 within a digital mixture 115 is activated in response to a request by an agent to access the digital mixture 115. In some embodiments, a super-identity is embedded in the ICTO 115 and includes criteria to verify an identity of an agent attempting to access the ICTO 115, dynamic rules to provide an intelligent awareness that validates the agent and determines the data's current state, and algorithms for data cloaking as specified in cloaking patterns. Verification criteria such as challenge/response pairs, digital signatures, biometric information, and/or the like may be used to verify the identity of the agent. At block 304, the portable dynamic rule set 108 is executed to verify that the agent is allowed the requested access to the digital mixture 115 in a relevant context. The identity module 109 and the intelligence module 111, when activated, assess the current access attempt by the verified agent and

establish a level of trust. In some embodiments, this assessment is an on-going process, in that there is a continuous verification and validation of each participant 101: the data owner, the agent (data user) and the data itself. In some embodiments, pre-access rules from the portable dynamic rule set 108 may be executed by the mixer 110 to decrypt at least some portion of the ICTO 115 for internal use by the mixer 110 without allowing access to the decrypted data to agents other than the mixer 110. Pre-access rules have access to the participants 101, including the ability to test identity artifacts and evaluate owner and agent data. If the trust level goes down, the protocol reassesses the participants 101. In some embodiments, if the agent attempting to access the ICTO 115 is unable to re-establish their legitimacy, defensive or offensive actions may be invoked. If the agent is able to satisfy the new set of challenges, access will be allowed to proceed or continue.

In some embodiments, the pre-access rules are merely allowed read access to identity data, but in some embodiments, the pre-access rules may also have write access, which may be used, for example, to record access attempt attributes when opening (or attempting to open) the ICTO 115.

The method 300 proceeds to block 306, where the portable dynamic rule set 108 determines a cloaking pattern used to protect the requested data. The portable dynamic rule set 108 consults the mixture metadata 104 to determine which cloaking pattern 107 was applied based on the identity of the agent, the data request, the context in which the data is being requested, and/or the like. Once the used cloaking pattern 107 is determined, the method 300 proceeds to block 308, where the cloaking pattern 107 is used to provide the requested access to the agent. Similar to how the cloaking pattern 107 indicated a set of techniques used to protect the requested data, the cloaking pattern 107 also indicates a set of techniques used to reconstruct the requested data from the protected version stored in the ICTO 115. The method 300 then proceeds to an end block and terminates.

FIGURE 4 is a schematic diagram that illustrates an exemplary use case for an embodiment of the present disclosure. One of ordinary skill in the art will recognize that this use case is exemplary only and is described to show certain features of the disclosure, but that this use case does not utilize or describe every feature of the technology disclosed herein. In FIGURE 4, a first user 418, using a first computing device 416, uses an embodiment of the present disclosure to protect a first piece of data (data one 404) and a

second piece of data (data two 406). An ICTO 408 is created that includes a protected version of data one 410 and a protected version of data two 412. In creating the ICTO 408, the first user 418 specifies that a second user 422 may access data one 404, but does not specify that the second user 422 may access data two 406. Hence, the
5 ICTO 408 includes a rule in its portable dynamic rule set 108 that allows user two 422, once verified, to access data one 404.

The first computing device 416 transmits the ICTO 408 to a second computing device 420 used by the second user 422 via a network, such as a LAN, a wireless network, the internet, and/or the like. The second user 422 causes the ICTO 408 to be
10 activated, and submits a request 424 to access to data one 404. The ICTO 408 verifies the identity of the second user 422, which may include processing a challenge/response pair stored in the ICTO 408 and/or consulting a trusted service 409 (such as a certificate server, a RADIUS or other authentication server, and/or the like) to verify that the second user 422 is who he purports to be. Once the identity of the second user 422 is verified,
15 the ICTO 408 consults the cloaking pattern used to create protected data one 410, and uses the cloaking pattern to give the second user 422 access to data one 404. The second user 422 may also submit a request 426 to access data two 406. However, because the ICTO 408 has not been instructed to provide access to data two 406 for the second user 422, the ICTO 408 does not allow the second user 422 to access data two 406.

20 Though a trusted service 409 that provides authentication services is described, other types of trusted services may be used. For example, if a rule is included the ICTO 408 that only allows access during a given time period, a trusted service 409 that provides a trusted date-time value may be used. As another example, a trusted service 409 may seek input from other users while the ICTO 408 is determining whether
25 to grant access to an agent. As illustrated, a trusted service 409 may notify the first user 418 of the access attempt via email, SMS, or any other suitable technique, and may wait to allow the attempted access until a corresponding approval is received from the first user 418.

This use case illustrates several advantages of the present disclosure. Once the
30 ICTO 408 is created, protected data one 410 and protected data two 412 cannot be accessed without invoking the processing of the ICTO 408 to request access. Accordingly, the data is protected when the ICTO 408 is stored on the first computing device 416, when the ICTO 408 is in transit on the network 402, and when the ICTO 408

is stored on the second computing device 416. Also, even though the ICTO 408 provides access to the second user 422 to data one 404, data two 406 is nevertheless protected from access.

While this simple use case illustrates several features of the present disclosure, much more complex use cases are also possible. For example, FIGURE 5 is a schematic diagram that illustrates aspects of an exemplary workflow for an embodiment of the present disclosure. A first user ("User A") may have a set of documents ("Documents X, Y, and Z") to be signed by a second user ("User B"), a third user ("User C"), and a fourth user ("User D"). Document X needs to be signed by User B. Document Y needs to be signed by User B and User C, but only after Document X has been signed. Document Z needs to be signed by User D, but only after Documents X and Y have been signed. Further, Document X and Document Y must be signed during working hours (between 9AM and 5PM) so that assistance may be provided if necessary, while Document Z must be signed before an expiration date.

Embodiments of the present disclosure will support such a workflow. User A creates an ICTO that includes Documents X, Y, and Z. User A creates an access rule for Document X that allows User B to review and sign Document X. User A creates an access rule for Document Y that allows User B and User C to review and sign Document Y once the signature on Document X is obtained. User A may create an access rule for Document X that allows User C to review Document X to check for a signature, or the access rule for Document X may detect the signature applied to Document X, and may dynamically update the access rule for Document Y that allows it to be signed once the signature is detected. User A creates an access rule for Document Z that checks for signatures on Documents X and Y, and upon detecting such signatures, User D is allowed to sign Document Z. Each of these rules also enforces the associated time requirements, and does not allow access if the time requirements are not satisfied. User A may also create a rule that reports any access to any of the documents back to User A, so that User A may monitor the process. Each of the rules may specify how each user is to be identified, the related privileges, devices from which the users are allowed to access the documents, and locations from which the users are allowed to access the documents.

Once, for example, User B receives the ICTO, User B invokes an application configured to activate the executable code within the ICTO. The executable code determines the identity of User B, either by consulting a trusted identity service, by

checking the response to a challenge included in a rule, or by any other method. Once the identity, time, location, and other requirements are satisfied, User B is allowed to access Document X, but not any of the other documents. After User B signs Document X, the ICTO is transferred to the next user, and enforces the protections on the documents as the
5 ICTO passes through the rest of the workflow.

One of ordinary skill in the art will recognize that the above use cases are exemplary only, and that many other use cases for the subject matter disclosed herein are possible. For example, because the portable dynamic rule sets include executable code, the ICTO may protect executable content that is only executable upon satisfying the
10 security checks of the ICTO. Also, since the ICTO may execute such content in response to the success or failure of any rule, the ICTO may log successful accesses or take action such as alerting a data owner, initiating a self-destruct sequence, or other actions upon detecting an unauthorized access attempt.

FIGURE 6 is a block diagram that illustrates an exemplary hardware architecture
15 of a computing device 500 suitable for use with embodiments of the present disclosure. Those of ordinary skill in the art and others will recognize that the computing device 500 may be any one of any number of currently available or yet to be developed devices including, but not limited to, desktop computers, server computers, laptop computers, embedded computing devices, application-specific integrated circuits (ASICs),
20 smartphones, tablet computers, and/or the like. In its most basic configuration, the computing device 500 includes at least one processor 502 and a system memory 504 connected by a communication bus 506. Depending on the exact configuration and type of device, the system memory 504 may be volatile or nonvolatile memory, such as read only memory ("ROM"), random access memory ("RAM"), EEPROM, flash memory, or
25 similar memory technology. Those of ordinary skill in the art and others will recognize that system memory 504 typically stores data and/or program modules that are immediately accessible to and/or currently being operated on by the processor 502. In this regard, the processor 502 serves as a computational center of the computing device 500 by supporting the execution of instructions.

30 As further illustrated in FIGURE 6, the computing device 500 may include a network interface 510 comprising one or more components for communicating with other devices over the network. Embodiments of the present disclosure may access basic services that utilize the network interface 510 to perform communications using common

network protocols. In the exemplary embodiment depicted in FIGURE 6, the computing device 500 also includes a storage medium 508. However, services may be accessed using a computing device that does not include means for persisting data to a local storage medium. Therefore, the storage medium 508 depicted in FIGURE 6 is represented with a dashed line to indicate that the storage medium 508 is optional. In any event, the storage medium 508 may be volatile or nonvolatile, removable or nonremovable, implemented using any technology capable of storing information such as, but not limited to, a hard drive, solid state drive, CD ROM, DVD, or other disk storage, magnetic cassettes, magnetic tape, magnetic disk storage, and the like.

As used herein, the term "computer readable media" includes volatile and nonvolatile and removable and nonremovable media implemented in any method or technology capable of storing information, such as computer readable instructions, data structures, program modules, or other data. In this regard, the system memory 504 and storage medium 508 depicted in FIGURE 6 are merely examples of computer readable media.

Suitable implementations of computing devices that include a processor 502, system memory 504, communication bus 506, storage medium 508, and network interface 510 are known and commercially available. For ease of illustration and because it is not important for an understanding of the claimed subject matter, FIGURE 6 does not show some of the typical components of many computing devices. In this regard, the computing device 500 may include input devices, such as a keyboard, mouse, microphone, touch input device, and/or the like. Similarly, the computing device 500 may also include output devices such as a display, speakers, printer, and/or the like. Since all these devices are well known in the art, they are not described further herein.

While illustrative embodiments have been illustrated and described, it will be appreciated that various changes can be made therein without departing from the spirit and scope of the claimed subject matter.

CLAIMS

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A computer-readable medium having an intelligent cipher transfer object stored thereon;

wherein the intelligent cipher transfer object comprises a set of participants including a portable dynamic rule set; and

wherein, in response to execution by one or more processors of a computing device, the portable dynamic rule set causes the computing device to perform actions comprising:

receiving, from an agent, a request for access to a portion of a participant of the set of participants;

verifying that the agent is authorized to access the requested participant portion; and

providing access to the requested participant portion for the agent without providing access to other portions of the set of participants for the agent.

2. The computer-readable medium of Claim 1, wherein the portable dynamic rule set is located at a variable location within the intelligent cipher transfer object, and wherein the portable dynamic rule set can be found within the intelligent cipher transfer object for execution based on an attribute of the intelligent cipher transfer object.

3. The computer-readable medium of any of Claims 1-2, wherein the portable dynamic rule set includes at least one rule that indicates an agent that may access a participant and contexts in which the agent may access the participant.

4. The computer-readable medium of Claim 3, wherein the contexts in which the agent may access the participant include one or more of a time period, a location, and an identity of a computing device.

5. The computer-readable medium of any of Claims 1-4, wherein the set of participants includes a set of cloaking patterns and mixture metadata, wherein each cloaking pattern of the set of cloaking patterns indicates a set of transformations applied to a participant in the intelligent cipher transfer object, and wherein the mixture metadata

includes information indicating which cloaking pattern of the set of cloaking patterns is usable to access each participant of the set of participants.

6. The computer-readable medium of Claim 5, wherein a first cloaking pattern is usable by a first agent to access a given participant, and wherein a second cloaking pattern different from the first cloaking pattern is usable by a second agent to access the given participant.

7. The computer-readable medium of Claim 5, wherein a first cloaking pattern is applied to a first participant, and wherein a second cloaking pattern different from the first cloaking pattern is applied to a second participant.

8. The computer-readable medium of Claim 5, wherein the set of transformations includes one or more of a data compression technique, a data normalization technique, and a data encryption technique.

9. The computer-readable medium of any of Claims 1-8, wherein the portable dynamic rule set causes the computing device to perform further actions comprising:

in response to determining that a request for access to a portion of a participant of the set of participants is not authorized, causing at least one rule in the portable dynamic rule set to be executed to handle the unauthorized access request.

10. The computer-readable medium of Claim 9, wherein the at least one rule to be executed to handle the unauthorized access request causes the intelligent cipher transfer object to self destruct.

11. The computer-readable medium of Claim 9, wherein the at least one rule to be executed to handle the unauthorized access request causes a message to be sent to a data owner.

12. The computer-readable medium of Claim 9, wherein the at least one rule to be executed to handle the unauthorized access request causes a record of the unauthorized access request to be stored in the intelligent cipher transfer object.

13. The computer-readable medium of any of Claims 1-12, wherein the portable dynamic rule set causes the computing device to perform further actions comprising:

in response to determining that a request for access to a portion of a participant of the set of participants is authorized, causing at least one rule in the portable dynamic rule set to be executed to handle the authorized access request.

14. The computer-readable medium of Claim 13, wherein the at least one rule to be executed to handle the authorized access request causes the requested portion of the participant to be uncloaked and provided to the requesting agent.

15. The computer-readable medium of Claim 13, wherein the at least one rule to be executed to handle the authorized access request causes a message to be sent to a data owner.

16. The computer-readable medium of Claim 13, wherein the at least one rule to be executed to handle the authorized access request causes a record of the authorized access request to be stored in the intelligent cipher transfer object.

17. The computer-readable medium of Claim 13, wherein the at least one rule to be executed to handle the authorized access request causes a signature of the requesting agent to be associated with the requested portion of the participant.

18. The computer-readable medium of Claim 13, wherein the at least one rule to be executed to handle the authorized access request causes at least one rule of the portable dynamic rule set to be added, modified, or deleted.

19. The computer-readable medium of any of Claims 1-18, wherein the intelligent cipher transfer object further comprises a nested intelligent cipher transfer object.

20. The computer-readable medium of any of Claims 1-19, wherein the intelligent cipher transfer object is nested within an outer intelligent cipher transfer object.

21. The computer-readable medium of any of Claims 1-20, wherein the portion of the participant of the set of participants is stored in the intelligent cipher transfer object in an encrypted state, and wherein providing access to the requested participant portion for the agent includes decrypting the encrypted portion of the participant of the set of participants using a decryption key without providing the decryption key to the agent.

22. A computer-implemented method of creating the intelligent cipher transfer object recited in any of Claims 1-21.

23. A computing device configured to execute the executable portions of the intelligent cipher transfer object recited in any of Claims 1-21 to provide access to the participants stored in the intelligent cipher transfer object in response to requests from authorized agents, and to otherwise prevent access to the participants stored in the intelligent cipher transfer object.

24. A computer-implemented method of protecting a set of participants, the method comprising:

obtaining, by a computing device, a set of participants to be protected;

determining one or more cloaking patterns for protecting the set of participants, wherein a first cloaking pattern is used to protect a first subset of the set of participants, and a second cloaking pattern different from the first cloaking pattern is used to protect a second subset of the set of participants;

applying, by the computing device, the determined cloaking patterns to the set of participants to create a set of cloaked participants, and

adding, by the computing device, the set of cloaked participants to an intelligent cipher transfer object.

25. The method of Claim 24, further comprising adding the one or more cloaking patterns to the intelligent cipher transfer object.

26. The method of any of Claims 24-25, wherein determining the one or more cloaking patterns includes determining a cloaking pattern for a given agent based on an identity of the given agent.

27. The method of any of Claims 24-26, wherein determining the one or more cloaking patterns includes adding an association to a mixture metadata participant indicating a cloaking pattern used to protect a given participant from access by any agent but a given agent.

28. The method of Claim 27, further comprising adding the mixture metadata participant to the intelligent cipher transfer object.

29. The method of any of Claims 24-28, further comprising adding a portable dynamic rule set to the intelligent cipher transfer object, wherein at least one rule of the portable dynamic rule set is executable by a computing device to manage access to the participants in the intelligent cipher transfer object.

30. The method of Claim 29, wherein adding the portable dynamic rule set to the intelligent cipher transfer object includes adding the portable dynamic rule set to a location in the intelligent cipher transfer object that is based on an attribute of the intelligent cipher transfer object.

31. The method of any of Claims 24-30, wherein the set of participants to be protected includes an intelligent cipher transfer object.

32. A computing device configured to perform the method of any of Claims 24-31.

33. A computer-readable medium having computer-executable instructions stored thereon that, in response to execution by one or more processors of a computing device, cause the computing device to perform the method of any of Claims 24-31.

34. A computer-implemented method of managing access to data protected by an intelligent cipher transfer object, the method comprising:

receiving, by a computing device, an access request from an agent to access a portion of a participant stored in the intelligent cipher transfer object;

activating, by the computing device, a portable dynamic rule set within the intelligent cipher transfer object;

executing, by the computing device, at least one rule in the portable dynamic rule set to evaluate the access request; and

in response to determining that the access request is permissible, providing access to the portion of the participant requested by the agent without providing access to other participant portions.

35. The method of Claim 34, wherein the access request from the agent includes a request to read the portion of the participant.

36. The method of any of Claims 34-35, wherein the access request from the agent includes a request to write to the portion of the participant.

37. The method of any of Claims 34-36, wherein the access request from the agent includes a request to add a signature associated with the portion of the participant.

38. The method of any of Claims 34-37, wherein activating the portable dynamic rule set includes determining a variable location of the portable dynamic rule set within the intelligent cipher transfer object based on an attribute of the intelligent cipher transfer object.

39. The method of any of Claims 34-38, wherein evaluating the access request includes verifying an identity of the agent.

40. The method of any of Claims 34-39, wherein evaluating the access request includes verifying that a location from which the access request was received is permissible.

41. The method of any of Claims 34-40, wherein evaluating the access request includes verifying that the access request was submitted during a permissible time period.

42. The method of any of Claims 34-41, wherein providing access to the portion of the participant requested by the agent includes:

determining a cloaking pattern used to protect the portion of the participant requested by the agent; and

using the cloaking pattern to obtain an accessible version of the portion of the participant from the intelligent cipher transfer object.

43. The method of Claim 42, wherein determining the cloaking pattern includes retrieving a cloaking pattern from a cloaking pattern participant of the intelligent cipher transfer object based on information associated with the agent and stored in a mixture metadata participant of the intelligent cipher transfer object.

44. The method of any of Claims 34-43, wherein evaluating the access request includes creating a record of the access request and storing the record in the intelligent cipher transfer object.

45. The method of any of Claims 34-44, wherein evaluating the access request includes sending a notification to a data owner regarding the access request.

46. The method of any of Claims 34-45, further comprising, in response to determining that the access request is not permissible:

denying access to the portion of the participant; and

executing at least one rule in the portable dynamic rule set for taking defensive action.

47. The method of Claim 46, wherein the defensive action includes causing the intelligent cipher transfer object to self-destruct.

48. The method of any of Claims 46-47, wherein the defensive action includes notifying a data owner of the impermissible access attempt.

49. A computing device configured to perform the method of any of Claims 34-48.

50. A computer-readable medium having computer-executable instructions stored thereon that, in response to execution by one or more processors of a computing device, cause the computing device to perform the method of any of Claims 34-48.