Improvements in Blind Encrypted Data Matching

P. L. Baker
WWN Software LLC

February 25, 2012

Table of Contents

BACKGROUND

Information Security and Information Sharing

Information security is a broad field that entails the protection of data from theft, alteration, or destruction. Technical measures to strengthen information security include enhancements to the physical security of the site where data resides, security software that protects the computer processes from viruses, worms, Trojans, etc., intrusion detection that alerts defenders when physical or software security is probed, and finally encryption which renders the data unusable unless an attacker can obtain the key that unlocks data encryption. In addition, information security can be divided into lifecycle phases: data at rest, data in motion, and data in use.

Information sharing is essential to the conduct of business and government. Business cannot reach a deal without negotiating over terms. Government cannot act effectively without information. Health care providers need information about patients to treat them successfully. Unfortunately, information sharing is in conflict with information security. When shared, information moves to a new location and is used by additional parties. Sharing exposes information to more points of attack from intruders.

Although information sharing and information security are fundamentally incompatible, it is possible to minimize the security risks and also maximize the value of sharing by adopting a process that negotiates the exchange of limited amounts of high value information without placing the bulk of the information at risk. This is the process of "Blind Encrypted Data Matching" or BEDM.

BEDM encrypts data while it is being searched for opportunities for information. Thus, it maintains information security for data up to the point when a decision is made to share selected, relevant data. The deployment of BEDM is urgently required in many applications because older information technology requires data to be decrypted at the point of use and such a decryption step creates an opportunity for data misuse. On first reading of the

preceding statement, it seems logical to object that data cannot be used while encrypted because encryption is designed specifically to conceal data and thus to prevent its use. In a broad sense, that may be true; however, there is a vitally important subcategory of data processing that involves assembling related records to permit an integrated analysis. BEDM offers a secure way to assemble related records by identifying related records while they remain encrypted. In operation, two or more parties conceal their data securely and negotiate intelligently over an agreement to share carefully selected information records.

Negotiation of Data Access

To illustrate the need for negotiated data access, we will consider the work in the Office of Director of National Intelligence (ODNI) as an example. The work of the ODNI is essential to national security but it can conflict with the public interest for personal data privacy. The ODNI must obtain and integrate vast amounts of data from every security agency and financial institution in order to recognize and respond to threats from terrorist groups. Of course, the ODNI provides state-of-the-art protection for the data-in-motion as it arrives at the central resource and it protects the data-at-rest. However, the current computer systems must still provide for on-the-fly decryption of the data before it can be correlated, analyzed, and combined into actionable intelligence. At this point, the data system is vulnerable to misuse whether such misuse is ideologically motivated, motivated by profit, or simply inadvertent. Moreover, all the agencies that contribute the data must place their intelligence-data sources at risk when the data are included in the national archive. A counterintelligence agent reporting to an outside power can identify the intelligence-data sources by inspecting the information obtained from a source. Then such an intruder can betray the source and silence it. That outcome is highly undesirable because it can take years to develop an intelligence-data source; therefore, the data source's handlers are naturally reluctant to share the source's reports in their entirety if the data will be widely available for analysis.

An important, albeit mostly unacknowledged, human factor in modern information sharing environments is the reluctance to work with others because of the fear of loss or betrayal. Fear inhibits organizational effectiveness. BEDM replaces conventional, unconditional,

uncontrolled data sharing with negotiated information sharing that limits the risk to people and programs and gives people a measure of control over sharing that they lack today. Over time, the fear of information sharing should subside.

In its most refined form, the negotiation takes place before there is any decryption and thereby information is protected during negotiation. Encrypted negotiation requires machine support but it would be a mistake to focus only on its technical properties. It is actually a machine-enabled extension of historically successful negotiation conditions into the domain of encrypted information. Historically, people negotiate face-to-face without revealing their full negotiation position, that is, without disclosing sensitive, concealed information. They reach agreement by a back-and-forth exchange of limited data and proposed agreement terms. In the new domain of computer-enabled negotiation over concealed terms, they can achieve similar ends with similar safety by supplying encrypted negotiation positions and allowing a machine to find a mutually agreeable match of terms without exposing any data to risk by decryption.

Networks of Computers and Security

Complex operations involve many users and many individual computers. The computers are interconnected with a network so that each computer unit can interact other units irrespective of physical location. Such computer units may comprise mainframe computers, rack-mounted computers, desktop computers, portable computers and personal-data-assistant (PDA) devices. In the following we describe such associations of computers operating together as a network of computers and we limit our discussion to processes that require more than one computer. The relationship of the computer units to each other characterizes the architecture of the network of computers. As with the architecture of buildings, networks of computers may adopt different architectures for different purposes. Unlike a building, however, a computer network may support several software architectures simultaneously. We begin our background review with the well-known client-server architecture, which characterizes each two-computer interaction as a client-server relationship. During each step in a process that is implemented with the client-server architecture, one unit acts as the client

and the other acts as the server. During other steps, of course, the units may play different roles. We will now use this simple client-server architecture to discuss information security and observe the strong connection with negotiation.

Figure 1 illustrates how information can be obtained from a data store located within a network of computers by means of computer units operating in the client-server architecture. In Unit 1 of Figure 1, the computer unit serves an analyst interacting in real-time or operates an algorithm specified by an analyst on a specified schedule. In either case, the analyst requires information that resides on another computer, for example, data residing in Data Store 3 on Unit 2 of Figure 1. To perform the process, Unit 1 - acting as a client - sends a data-message requesting something from Unit 2 – acting as a server – and Unit 2 responds by sending information from Data Store 3 to Unit 1. Thus, the client-server relationship is a simple sequence: the client (Unit 1) initiates and then the server (Unit 2) responds. When the information is sensitive, proprietary, or valuable, the information must be protected as illustrated in Figure 1. First, information in Data Store 3 is encrypted in the storage unit by encryption process 5. Therefore, the data are protected from hostile attacks on storage unit 3. Second, information passing between Units 1 and 2 is encrypted by encryption process 4 so that "data in transit" is secure. Data in transit is protected by encryption software installed on both computers 1 and 2 and a temporary encryption key determined when the connection between 1 and 2 is established in accordance with well-known protocols. Therefore the data are protected from hostile attacks on the network infrastructure of the computer network. It is clear, however, that the data in Data Store 4 is not protected in any way from the hostile operation of an analyst or algorithm operating in Unit 1. The server – Unit 2 in the example of Figure 1 – has no control of what purpose it may aid or abet on the client – Unit 1. In many practical situations, this is an untenable outcome for the server.

To see how the client-server relationship is untenable for certain purposes we will consider classified data recalling the ODNI example above. In fact, ODNI does operate a very large data storage facility containing the "crown jewels" of the intelligence agencies' operation. Clearly, if an unauthorized person could use the client to access data on such a server, then classified information could be disclosed contrary to law and against the interests of national

security. Therefore, a server containing classified data must conduct a negotiation with the client that entails, at the very least, the question “who is asking for service”. Such a negotiation is usually completed in a short time-period at the beginning of a series of client-server interactions by means of a so-called “login procedure” which identifies the human or institutional identity that is responsible for the interactive session or algorithmic process which initiates the client server interaction. For classified information, the “login procedure” is insufficient because a person may have a valid login but use the login identity for a hostile purpose. A server computer cannot simply offer all its data for access even if the identity behind the client’s request is proven. Consequently, any realistic network of computers in operation today will include some kind of “negotiation” over access rights. Typically, each client identity is assigned “access rights” on some basis. Each unit of data stored by the server units is assigned “access requirements”. With such assignments in place, a server unit must analyze each client request and match access rights against the access requirement on the basis of access rules before allowing data to pass from server to client. Such systems are state-of-art but suffer two limitations: first, the implications of the access rules are hard to anticipate leaving room for unintended data breaches; and second, the client’s purpose is revealed to the server by the specific requests that a client makes of the server. Hostile agents collocated at the server may gain access to the client queries, infer the intent of the investigation and disclose the client’s investigation without authorization.

We have discussed the client-server architecture above because it is common and familiar; moreover, it illustrates important security related issues. The center of our attention, however, is the negotiation-agent architecture in which one computer unit is the agent and all others operate as equals; that is, as peers. Individuals or administrative groups operate the peer computers. These operators can act independently, they may have diverging interest areas, they may be competitors, and they may be untrustworthy friends. Each peer can communicate directly with another peer computer on the network but peers may refuse to interact directly without negotiating a prior agreement about what information should be exchanged or revealed. That is where the negotiation agent comes in. The negotiation agent works with the peers to negotiate agreements that define the eventual peer-peer interactions. Examples include social networking sites (e.g. LinkedIn or Facebook) where the peers give

the agent sufficient information such as school graduation years or previous employers and the agent then makes suggestions to each peer about what other peers it may wish to connect with. Another example is a price-consolidation site that receives requirements from peers who desire to buy an airline ticket and then the agent matches the buyer peer with a seller who offers a good price within the buyer's terms (e.g. PriceLine or Kayak). A third example is a fraud alert service in which an agent inspects transactions at several peer-level financial sites and alerts the sites when there are patterns of activity that may indicate illegal activity. This third example is interesting because it is a persistent activity; that is, the agent runs constantly to monitor ongoing activity and look for a special kind of pattern. This example illustrates another property of negotiation agents: they can operate on their own schedule on behalf of peer-level computers who are free to mind their own business until the agent issues an alert.

Figure 2 illustrates the negotiation agent architecture. The peer-level computer units are labeled (2) in the figure and represent independent interests or entities. The peers can communicate directly via network communication connections illustrated by (3) in Figure 1. However, for security and other reasons, they may require the services of the negotiation agent (1). The agent negotiates the terms of a future connection (3) so that the interests of both peers are well represented during the peer-to-peer interaction. Each peer (2) maintains information called local peer data. A peer will share some of this information with the negotiation agent during negotiation. The peer may also create queries and requests that it wants to satisfy. These are also shared with the agent. The negotiation agent (1) assembles the local data from the peers, aggregates all the queries from the peers and then infers the best response that is delivered back to each peer (2).

An undesired outcome of the negotiation agent architecture is that the agent gains a synoptic view of all the data, all the queries, and all the activities of the peers. From a security standpoint, such a consolidation of sensitive data is dangerous. Consequently a few implementations of this architecture have added a technical improvement- data encryption with an encryption key shared by the peers but not by the agent. The key is limited to the peers in a particular group as illustrated in Figure 2 by the region (4). The negotiation agent

(1) lies outside the domain. It does not possess the encryption key; therefore, it cannot encrypt the negotiation terms of the peers in the domain. When the agent finds results, the results are also encrypted with the key that it does not posses. Such a technical improvement greatly improves security but it limits the work of the negotiation agent to work that can it can perform on concealed terms.

Cloud Computing and Security

According to the Computer Security Division of NIST (csrc.nist.gov), "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". Cloud computer represents an application of a network of computers to provide flexible services emulating a variable set of virtual computer units. It is important to mention cloud computing in our background discussion because the cloud-computing concept may be employed in the implementation of any of the computer based services discussed here. The adoption of cloud computing should not add any complexity to the discussion except for the following observation. The fundamental basis of computer security is physical security; because software locks and encryption are ineffective if an intruder gains access to the actual computer hardware during an attack. It follows that cloud computing complicates security because there is no longer a fixed layout of physical hardware units instead it is necessary to defend a constantly variable configuration and perimeter.

Negotiation with Concealed Terms

Early approaches for negotiation over concealed terms were based on the observation that information records can be matched even if they are encrypted because of the following fairly obvious property: field values in the information record that match by equality before encryption will match by equality after the values are encrypted if and only if the same encryption key and encryption algorithm are used for all records. In the past, it was also common to adopt a one-way encryption algorithm. One-way encryption is conventionally

implemented as a hash function, such as SHA1, supplemented by a "salt" that serves as the encryption key. If an intruder obtains such encrypted data, the one-way encryption operation cannot be reversed by an inverse algorithm even if the intruder also has the key. To use this property for negotiation over concealed terms, the parties agree on a key or "salt" and then perform one-way encryption on the records before sending them to a negotiation agent for matching. In this simple architecture, the key receives no special care because advocates of one-way encryption believe that this application of encryption is irreversible. Negotiation systems built around one-way encryption are available commercially (e.g. IBM's Anonymous Resolution) and have been deployed in several federal agencies; however there are serious drawbacks that impede their widespread use.

The first disadvantage is that if the key is discovered there is an easy way to decipher a field that has been encrypted by a one-way method. Although there is no inverse for the one-way encryption algorithm, the data is still vulnerable to a code-dictionary attack. The attack starts with a list of words obtained from the dictionary, supplemented by lists of names, supplemented by numbers and dates, etc. The attacker encrypts each item on the list to obtain a code dictionary that allows two-way mapping between a clear text symbol and its corresponding encrypted symbol. The attacker then uses the code dictionary to decipher each field of each record and defeat the encryption.

A second disadvantage is that sophisticated analysis of the information will require additional matching operations beyond the single equality test. For example, many prospective applications require a way to match an encrypted number to a range of numerical values. Such range matching is needed to support searches on biometric measurements, which have some degree of uncertainty due to measurement errors or minor changes in the subject's appearance. Likewise, range matching would permit comparing descriptions of locations from witnesses who may be roughly accurate about location with locations derived by precise GPS-enabled surveillance devices. Moreover, business negotiations require matching high-low price ranges and minimum-maximum tolerance specifications.

The third disadvantage is conceptual, but it severely limits the acceptance of first generation systems in any enterprise. Whether the data are encrypted or not, the parties who supply the data give up all control. They have only one choice: send all of the data to a central location where all the data can be used by all authorized parties. In a state-of-the-art data security system, such as that used at ODNI, individual data items have detailed access privileges to limit access; but, from the data provider's standpoint, the provider loses control once the data leaves the home facility. Historically, data providers had more control and resist the loss of control.

Before computer-based negotiation agents can gain acceptance, they must convince users that they are satisfactory replacements for familiar, established practice. In established negotiation practice each party can regulate information sharing based on the reliability and trustworthiness of the opposite party. In the past, people regulated information sharing by negotiating face-to-face. The natural attitude was "help me meet my objective and I will help you with yours" and the fear in the background was always "this data can be used to damage me, my reputation, and my organization's goals; how can I entrust it to anonymous parties?" For the people involved, the data system users, these attitudes have not changed. Consequently, it has been difficult to convince people to adopt the early negotiation systems given that such systems do not allow fine-grained control of who may use the data.

Another approach was introduced that is described in U.S. Patent US 7,685,073 B2 entitled "Method and Apparatus for Negotiating Agreement Over Concealed Terms Through a Blind Agent". For conciseness, we refer to this method as "Blind Encrypted Data Matching" or hereafter BEDM. The process disclosed by the patent employs a blind agent that matches terms submitted by one party with terms submitted by another party when all terms have been concealed by an encryption algorithm and a short-lived, session encryption key that is specific to the temporal session and to a selected group of participants in the matching process. When members of the group provide the encrypted data for matching purposes, they can know with confidence that it will be matched only with selected parties that are familiar to them. In addition, each party can specifically tailor the information for negotiation bearing in mind which other parties are part of the negotiation.

The process for BEDM described in the previously cited patent specifies a procedure that carefully protects the key so that no party has the key and also has access to all the data. Consequently the blind agent creates a wall that prevents one party from browsing through the data of another. Secondly, the patent describes how partial encryption may be used to allow flexible, sophisticated matching guided by certain general words that remain unencrypted to preserve textual or data structure syntax.

In addition, the blind agent of the patent advertises the matching algorithms that are used so that each member of the group can be assured that the purposes are fair, legal, and consistent with goals. Lastly, the patent describes a novel method of concealing data with the encryption key that allows range matching. Thus, the BEDM computer-enabled process described in the patent fixes the major problems of the earlier approach.

Partial Encryption of Negotiation Terms

Negotiation of agreement over concealed terms through a blind agent produces, when possible, an agreement that allows the participating parties to exchange specific, concealed information records for their mutual benefit.

The agent is called “blind” because it receives only partially-encrypted information from the parties and the blind agent cannot learn anything useful from this encrypted information. The phrase “partially-encrypted” denotes that certain common words in sentences and statements are left unencrypted so that the syntactic pattern of the statement remains clear. Typically, nouns, adjectives and adverbs would be encrypted while verbs and conjunctions would not be. For statements written in a computer language, the language will define certain reserved words such as “or”, “and”, “select”, “where”, etc. In such languages, the reserved words are unencrypted and all other symbols in the statement are encrypted.

In contemporary systems, the preferred syntax of statements for information sharing is the XML notation system. For the context of XML, the process of partial encryption is

implemented by encrypting all symbols except for the XML tags and attribute names within XML tags.

AREAS FOR IMPROVEMENT

As with any invented process, experience reveals aspects where innovations can improve the process of negotiation over concealed terms. First, the growing use of cloud computing complicates information security at the facility that operates the blind-agent, i.e. the negotiation agent. Second, negotiations can be broken or disputed; therefore, the process must allow for adjudication of disputes. Third, there is room for improvement in the mechanism for concealing numerical value ranges.

The BEDM system is more secure than any earlier systems but some vulnerable points can be found. If there is a single defector in the whole enterprise, then the defector can only damage the site where the defector has access. BEDM cannot limit damage at one site, that is the job of conventional site security; BEDM can only prevent trouble from propagating to other sites. But suppose there are two defectors: one who works for a party that participates in the negotiated data matching and another who works for the site that operates the blind agent. If the first defector can corrupt the local system and get access to the session keys, then the session key can leave the secure BEDM system. Now if the second defector can gain access to the encrypted data that the blind agent receives for matching purposes, the second defector can export the encrypted data. Finally, a hostile third party who manages the two defectors receives both the data and the key that encrypts it. Thus a two-defector attack organized from the outside can compromise the BEDM system. For each party in the BEDM information-sharing negotiation, site security does not present any new problems when BEDM is introduced. However, the blind-agent is a new party at a new site and it is special to the BEDM system. Therefore, the site security for the blind-agent is a new problem and worthy of close scrutiny.

The blind agent is automated so there are very few operators working on site who might introduce covert channels or software viruses at the blind agent's site; therefore, site security is not difficult for a small site. However, the two-defector attack is a matter of concern for very large organizations because of cloud or farm computing. When a large organization matches data, the amount of data is so large that the matching process must be split out to a

cloud of computers or a local farm of computers working in parallel. Cloud computing increases total throughput but multiplies the number of computers that can be attacked by an intruder. Thus, very large organizations need a new security measure to protect the cloud of computers that hosts the blind-agent function.

The second area for improvement is the adjudication of disputes. The BEDM data protection is so strong it complicates dispute resolution. To see the problem, suppose one of the parties participates in a negotiated agreement to exchange concealed data. As a result, that party receives matching information from another party. Suppose further that one party then misuses the data acquired from the other party. If data misuse occurs, an injured party has no sure means of redress. At best, the injured party can avoid future problems by refusing future cooperation with the abuser. There is a record of the negotiated exchange, but that record is encrypted. It could not be entered in evidence in arbitration or in court. The accuser can decrypt the agreement but a judge or adjudicator cannot verify its authenticity.

The third area for improvement is the method for concealing numerical values or ranges while preserving their range matching operations. Previous methods used a linear transformation with coefficients generated by encryption. The method conceals each value but the statistics are not as secure. Suppose an intruder can obtain a large number of values for a field such as age. The intruder may know what the mean age of the expected demographic should be. The average age of the records that the intruder has stolen is an estimate of that expected mean and this estimate may be usefully accurate if the stolen sample size is large. Therefore, the intruder can use the expected mean and the estimated mean to derive values for the scaling coefficients of the linear transformation. While this scenario is not a very practical one, an improved method is desirable in order to eliminate security concerns.

DESCRIPTION OF THE IMPROVEMENTS

Security with Two Session Keys

Overview of the Operation

Negotiation of agreement over concealed terms through a blind agent produces, when possible, an agreement that allows the participating parties to exchange specific, concealed information records for their mutual benefit.

The agent is called “blind” because it receives only partially-encrypted information from the parties and the blind agent cannot learn anything useful from this encrypted information. The phrase “partially-encrypted” denotes that certain common words in sentences and statements are left unencrypted so that the syntactic pattern of the statement remains clear. Typically, nouns, adjectives and adverbs would be encrypted while verbs and conjunctions would not be. For statements written in a computer language, the language will define certain reserved words such as “or”, “and”, “select”, “where”, etc. In such languages, the reserved words are unencrypted and all other symbols in the statement are encrypted.

In contemporary systems, the preferred syntax of statements for information sharing is the XML notation system. For the context of XML, the process of partial encryption is implemented by encrypting all symbols except for the XML tags and attribute names within XML tags.

The blind agent is capable of operating fairly sophisticated matching algorithms owing to the fact that it can recognize syntax in the partially encrypted statements. Blind matching algorithms rely on the fact that encrypted values are equal if and only if their unencrypted values are equal. In addition, methods exist to conceal values so that their concealed values can be compared for numerical order. We now consider why innovative new methods are needed to maintain strict security when the process is scaled upwards to a large number of parties and large collections of information.

To show the utility of the required enhancement, one should consider the typical internal implementation of the blind agent’s computer system as illustrated in Figure 3. In this typical implementation, simplified and idealized for three clients for the purpose of discussion, we

see that partially encrypted documents arrive at the blind-agent's receiving area denoted as component 1 in Figure 3. As illustrated, the receiving area (1) contains three partially encrypted documents.

The blind agent must then match documents in pairs by means of three matching processes labeled as component 2 in Figure 3. Given N clients, there are (N * (N-1))/2 matching steps in component 2. As the size of each document is large, the number and duration of the matching steps creates a scaling problem for the blind agent; namely, the more the service is used by more clients, the harder it is to finish the blind matching process in a timely fashion.

To complete the description of the typical internal implementation, we note that the results of the matching steps are accumulated in component 3 and written temporarily as a complete, partially-encrypted match result in document 4. In a typical implementation, the blind agent communicates a match result to a client if and only if that client owns one of the records referenced in the match. Consequently, the blind agent operates component 5 which creates customized match results for each client where each customized match-results document (component 6 of Figure 3) contains only matches that refer to a record owned by the client.

As the load on the blind agent increases, it will be driven towards a solution relying on large number of computing machines operated in a parallel configuration. This hardware solution is variously known as a computer farm or a cloud computing environment. Henceforth, we will refer to the hardware solution as cloud computing.

Improved Operation with Two Session Keys

The introduction of cloud computing reduces the security of the overall enterprise computing facility because the work and the data are now distributed over many computers. Each computer offers a point of attack by a hostile party. For this reason, a new technical enhancement is needed for the blind agent matching method to provide additional security for the cloud. The enhancement is explained in Figure 4.

The enhancement divides the blind-agent matching process between two configurations of components. Components within the embracing component 7 constitute a protection layer for the remaining components. Component 7 is essentially a security barrier like a network firewall; hence we will refer to Component 7 as the "cloudwall". The function of the cloudwall is to add security to each partially-encrypted document sent to the blind-agent negotiator. The additional security is provided by a second partial encryption step using a session key known only to the cloudwall. The documents, now twice partially-encrypted, are then forwarded to computer systems in the configuration labeled as component 8. Component 8 operates the same steps as those shown in Figure 3. The work of the steps in component 8 can be distributed to a large number of host computers to increase information throughput and reduce latency. We shall follow current nomenclature and refer to component 8 simply as the "cloud". Although there are many computers operating in component 8, and an intruder who enters any one of them cannot use the information that may be stolen. Information in the cloud is protected with 2 keys, one held by component 7 and one held jointly by the parties that are negotiating. Thus, the presence of 2 keys vastly complicates any potential attack on the information security.

To complete the operation, the enhanced system in Figure 2 returns the doubly encrypted match documents (6) to the cloudwall (7) where one layer of encryption is removed by processes (11) producing the singly encrypted documents (12) that are returned to clients of the blind agent. Note that Figure 1 and Figure 2 describe the same input and output functions; therefore, documents (12) in Figure 2 will be identical to documents (6) in Figure 1 if and only if the input documents (1) in Figure 1 are identical to the input documents (9) in Figure 2. The process in Figure 2 has higher throughput and reduced latency due to the hardware capability of the cloud computer configuration (8).

Enabling Adjudication of Disputes

Operation without the Improvement for Adjudication

There is a need for a process to adjudicate disputes that may arise over agreements negotiated through a blind agent over concealed terms. To illustrate the need, let us consider the sale of a machine by one party (the seller) to another (the buyer). During negotiation, the seller conceals seller's negotiation position, namely the inventory of machines for sale, their specifications, and prices. The buyer has similarly concealed the purchase flexibility, namely acceptable specifications and acceptable price range. A blind agent finds a match but all information is encrypted during the negotiation and the match itself is encrypted. Buyer and seller decode the match found by the blind agent and conclude a deal. This system is useful if buyer and seller are reliable. Suppose however that the seller lied about the specifications of the machine and the machine is unsatisfactory for the buyer. Likewise, the buyer may receive the machine but pay less than the agreed price to the seller. In either case there is a breach of contract. There is no way however to adjudicate an allegation of breach of contract. Buyer and seller each possess an unencrypted copy of the terms of agreement but there is no independent third party that can verify that either copy is valid.

It is a simple matter for the blind agent to retain a copy of the negotiated agreement but the terms of agreement are encrypted. Therefore the blind agent cannot serve as a witness to the negotiation as might a human intermediary.

Operation with a Process to Adjudicate Disputes

From analysis of the unimproved process, it is clear there is a need for an adjudicator who can resolve the disputes that arise when a blind agent negotiates an agreement over concealed terms. The current improvement introduces an adjudicator process running on an independent computer facility. The adjudicator process validates the veracity of a copy of the disputed agreement and thereby provides the evidence necessary to resolve the dispute by mediation or judicial remedy.

The adjudicator is a computer system operated by a party that is independent of the buyer, seller, and blind agent. The adjudicator's independence is necessary because the dispute resolution will cause the decryption of the encrypted agreement and thus break the complete security of the original encryption. It is essential that the adjudicator remain separated from the blind-agent so that the adjudicator cannot perform an unauthorized decryption of any additional information. The adjudicator does not see any of original encrypted data - in contrast to the blind agent. Moreover, the adjudicator will only see an encrypted agreement document if two conditions are met. First, one of the parties to an agreement must lodge a complaint and provide a reference to the agreement. Second, the blind agent must agree to provide he adjudicator with an encrypted copy of the referenced agreement. Thus, the adjudicator is highly constrained. However, within the imposed limits, the adjudicator can perform an important verification function for dispute resolution.

The computer-based process for adjudication is shown as an event-sequence diagram in Figure 5. The sequence of events in this process starts following the successful conclusion of a negotiation over concealed terms mediated by the blind agent negotiator. The blind agent negotiator has retained a copy of the basis for agreement, but it is encrypted. The time-sequence line for the blind agent negotiator is the rightmost vertical line in Figure 5.

One of the parties in the negotiation claims that another party negotiated in bad faith. The injured party desires to pursue an action against the opposite party but cannot prove the

existence of an agreement. The events initiated by that party are organized along the leftmost vertical line in Figure 5 labeled the “plaintiff”.

The improvement over existing practice is the introduction of a new party to adjudicate disputes. This party is called the “adjudicator” and events for the adjudicator occur along its time-sequence line, which is the middle vertical line in Figure 5.

The process is initiated by the plaintiff and concludes with the verification or denial of the compliant by the adjudicator. The steps in the process are the following:

(1) The injured party, hereafter the plaintiff, files a complaint against the other party with the adjudicator. The time-sequence line for the plaintiff is the leftmost vertical line in Figure 5.
(2) If the adjudicator accepts the complaint, the adjudicator notifies the blind agent by forwarding the complaint plus a request for the relevant, encrypted basis for agreement.
(3) The adjudicator requests that the plaintiff provide a copy of the session key for the agreement. (Note, the plaintiff may include the session key with the original complaint. If so, the session key is not forwarded from the adjudicator to the blind agent).
(4) Blind agent returns the desired encrypted basis for agreement that is in dispute between the two parties.
(5) Plaintiff provides the session key to the adjudicator if it was not provided earlier.
(6) Adjudicator uses the session key to decrypt the encrypted agreement.
(7) Adjudicator evaluates the plaintiff’s complaint in the context of the decrypted agreement and decides if the plaintiff is entitled to remedy.
(8) Adjudicator renders judgment on the complaint and notifies the plaintiff and the blind agent about the judgment.

Notice that the party who is the subject of the complaint plays no role in this process. The exclusion of one party is reasonable in this case because the process is not adversarial. The adjudicator is only asked to pass judgment on the evidence stored in the blind agent’s files and to certify the validity of the evidence. The implications of the evidence in a dispute between two parties must be handled in legal proceedings according to applicable laws.

Improved Method to Conceal Continuous Variables

Operation without the Improvement for Concealment of Continuous Variables

According to current practice, real numbers and real ranges are concealed by the application of a linear transformation function. The parameters of this function are determined from the assigned name of the number, the encryption key, and the encryption algorithm. However complex this determination may be to implement, the resulting transformation is simple. Therein lies a weakness in the method. To illustrate the weakness, we computed 10,000 random numbers having a Gaussian distribution characterized by a mean value of 50 and a standard deviation of 15. Then we follow current practice and map the values to new, concealed values using a linear function. The 10,000 concealed numbers were then accumulated in the histogram that is shown in Figure 6. This histogram clearly displays the original Gaussian distribution of the numbers.

The results shown in Figure 6 illustrate a weakness with respect to concealment. Suppose an attacker who wishes to compromise the security of the concealment is able to steal a large sample of the concealed values. Suppose also that the attacker has knowledge of the expected mean and standard deviation. Then if the attacker calculates the mean and standard deviation of the stolen values, it is straightforward to recover the coefficients of the linear transformation with considerable accuracy.

The weakness exposed in Figure 6 applies only for variables that follow a well-known distribution; many variables will not have known statistical behavior. Furthermore, the linear transformation for each named value is different and unique. Therefore information learned about one named value does not compromise another. Nevertheless the concealment system is weaker than it needs to be.

An Improved Method for Concealment of Continuous Variables

The method for concealing numerical values with an encryption procedure works in the context of a negotiation system satisfying the following assumptions:

Assumptions:
(1) the encryption algorithm has been selected and agreed upon by the parties
(2) there is an encryption key that is valid for the duration of the negotiation session
(3) the parties have agreed on a divisor D which divides the block length L of the encryption algorithm and which is considerably less than L; for example, $D < L/16$. The divisor will be used in the following as the number of linear segments in a piecewise linear function.
(4) the parties agree on the bit order in the binary computer representation of integer numbers (that is big-endian versus little-endian)
(5) the parties have determined and agreed upon a minimum value cmin that is the most negative number that can be represented on the least capable computer operated by the parties and a corresponding cmax which is the most positive number
(6) the parties have determined and agreed upon the allowed ranges of all numerical values which means that for each named value there is an associated minimum value, vmin[name], and an associated maximum value, vmax[name].
(7) the parties have determined and agreed upon a reduced representation range cdelta that is smaller than (cmax - cmin). Furthermore, cdelta should be sufficiently small that its value can be represented on the least capable machine.
The new, improved method is implemented in the context of these assumptions by construction and application of a piecewise linear function, PWLF, which is derived using from the name of the value to be concealed and the encryption key. The PWLF is then applied to all values associated with the name. The preferred method for constructing the piecewise linear function is as follows:

Steps:
(1) Compute the value vdelta= (vmax / D) - (vmin / D)
(2) Encrypt the name using the encryption key producing a string of bytes, denoted as B having a length of L bytes. In the uncommon case that the name string is so long that its

encrypted value is a byte string of a length which is larger than L, then the first L bytes of the encrypted value shall be used for B.

(3) The string of bytes B is divided into D substrings denoted by B[k] whose lengths sum to L where k = 1, 2, ... D. Note that B[k] denotes an array with an index k. This notation convention using square brackets will be used in the following without further explanation.

(4) Each substring B[k] is cast to a positive integer value I[k] using the agreed bit-order convention.

(5) Compute a scale factor A as follows:

S = summation of I[k] for k = 1, 2, ... D

A = (cmax - cdelta)/S - cmin / S ; whereby we strongly suggest the order of evaluation shown to prevent overflow of any machine register.

(6) The desired PWLF for the given name is defined by the D + 1 points of a sequence of (x, y) pairs which specify the value of y = PWLF(x) at the endpoint of each linear segment. The first pair of the sequence is designated by the index 0 and it is (x[0], y[0]) = (vmin, cmin). For any other pair in the range of k = 1, 2, ... D, the pair is (x[k], y[k]) where

x[k] = vmin + k * vdelta

y[k] = y[k-1] + A * I[k] + k * cdelta / D

It is apparent that the PWFL of x is monotonically increasing in y as x increases. Furthermore, there is a minimum slope which is established by the term containing the factor cdelta in the equation for y[k]. The minimum slope should be sufficient to ensure that the function is invertible within acceptable numerical accuracy. A sufficient slope is ensured by ensuring that cdelta is sufficiently large.

Discussion of the Improved Method for Concealment of Continuous Variables

The improvement consists in the substitution of a function with more parameters. A linear transformation has only two parameters. Predictable distribution functions such as the Gaussian function typically have two characteristic parameters. Therefore, from an attacker's viewpoint, the concealed data points can, in principle, estimate the unknown parameters of the linear transformation. That creates a weakness in the method currently in use. The

improved method substitutes a function of many parameters so that the parameters cannot be obtained by sampling the output of the function. Obviously, if an attacker were permitted to test the function with known inputs, then it would be simple to determine the function's parameters. However, this concealed variable method is intended for use in the context of a blind agent negotiation process that has safeguards to prevent isolation of the function and probing the function with known inputs.

The preferred method explained above uses a piecewise linear function. The concealment is most effective if the input data range uses all the available range of the function. To see why this is true, consider an input range that is very small compared to the range of the function. The range of the input would then lie entirely within one segment and the improved method is no better than the unimproved method. Therefore, the improved method advises the implementors to specify the range of each named variable (vmin and vmax), as explained in Assumption 6 in the preceding.

The next consideration is that the output of the function must be numerically acceptable to the numerical range and precision capacities present on all machines in an enterprise. If numbers on one machine have a smaller range than on another machine, it may be impossible for all machines to compute the concealed value for the eventual comparison with concealed values from the other machines. Consequently, Assumption 5 in the preceding advised that all parties in the negotiation must agree on a representation range (cmin, cmax) which is feasible for the machines operated by all parties.

Another consideration is numerical accuracy. The piecewise linear transformation is designed to increase monotonically to preserve the order of the values but we also need to ensure that values are distinguishable. Let us consider the difference of two numbers x and y which is
(x - y). When x and y are transformed to x'and y' with a monotonically increasing function it is certain that (x-y) and (x'-y') share the same sign. But if the monotonically increasing function is poorly chosen, the magnitude of (x'-y') may be insufficient to ensure numerical precision. This difficulty is avoided by insisting that slope of the function should have a

minimum value. In our formulation of the explanation, the minimum slope is defined by the parameter cdelta as discussed in Assumption 7 above.

Example of the Improved Process in Use

Let us consider the following non-limiting example of a specific embodiment to illustrate the operation of the improved method.

Let us assume the following:

- The encryption is AES 128 and the encryption key is "3c878d410d3c28af0285c9ff6467b8b5".
- The name of the value is "foobar".
- The piecewise linear transformation shall have 16 segments
- The binary representation of integer numbers is big-endian.
- The range of the values is (vmin, vmax) = (0.0, 100.0)
- The acceptable range of concealed values is (cmin, cmax) = (-1.0E+100, +1.0E+100)
- The reduced representation range (which creates a minimum slope) is cdelta = 0.5E+100, that is, 25% of the full, acceptable range.

Given this starting point, we follow the steps described above. We find in Step 2 that our encryption algorithm has a block size of 128 bits or L = 16 bytes. The result of encrypting the name "foobar" is the string of bytes: "d1791cb2d27e42fdb018f4a23bb17765" in hexadecimal notation.

We next follow Step 3 and divide the encrypted name into 16 substrings = d1, 79, 1c, b2, d2, 7e, 42, fd, b0, 18, f4, a2, 3b, b1, 77, and 65.

We next follow Step 4 and convert the substrings to integer values: 209, 121, 28, 178, 210, 126, 66, 253, 176, 24, 244, 162, 59, 177, 119, and 101.

Finally, we complete Steps 5 and 6 to produce the points that define the piecewise linear transformation by defining the endpoint so the linear segments. The values of x[k] are computed by the trivial iterator given in Step 6. The values of y[k] are as follows:

y[0] = -1.00000e+100
y[1] = -8.29602e+99
y[2] = -7.17793e+99
y[3] = -6.67901e+99
y[4] = -5.18142e+99
y[5] = -3.47079e+99
y[6] = -2.31941e+99
y[7] = -1.56749e+99
y[8] = 4.29427e+98
y[9] = 1.91370e+99
y[10] = 2.38599e+99
y[11] = 4.32299e+99
y[12] = 5.71405e+99
y[13] = 6.41936e+99
y[14] = 7.91029e+99
y[15] = 9.01506e+99
y[16] = 1.00000e+100

The preceding piecewise linear function was applied as a test to the original 10,000 points used to produce the illustration of Figure 6. The new, transformed points where also accumulated in the histogram which is shown in Figure 7. Notice how the distribution in Figure 7 is no longer Gaussian. Therefore, the improved method prevents using statistical methods to estimate the concealed value with any accuracy.

Although specific embodiments of the improvement have been described herein, it is understood by those skilled in the art that many other modifications and embodiments of the

improvement will come to mind to which the improvement pertains, having benefit of the teaching presented in the foregoing description and associated drawings.

SUMMARY

The process of negotiating over concealed terms and reaching an agreement incorporates a second session key in machinery of the blind agent (the broker for the negotiation) creating a security wall to cryptographically defend cloud computing resources. Any broken agreement can be adjudicated a new subprocess that can render judgment on disputes over the concealed terms of a previously negotiated agreement. Finally, the concealment of real numbers and numerical ranges -- a capability necessary for the quantitative specifications included in the proposed terms of agreement -- is greatly improved by the implementation of a piecewise linear transformation derived from the session's encryption key.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 – Client Computer and Server Computer in a Network of Computers

Figure 2 – Peer Computers using a Negotiation-Agent Computer in a Network of Computers.

Figure 3 - Typical Internal Implementation of the Unimproved Process for Negotiation of Agreement Over Concealed Terms through a Blind Agent

Figure 4 - Preferred Implementation of the Process for Negotiation of Agreement Improved with Two Session Keys

Figure 5 - Event Diagram for the Implementation of Adjudication of a Dispute Resulting from Negotiation of Agreement Over Concealed Terms through a Blind Agent

Figure 6 - Sample Histogram of 10,000 Random Numbers with Gaussian Distribution After Application of a Linear Function

Figure 7 - Sample Histogram of 10,000 Random Numbers with Gaussian Distribution After Application of a Piecewise Linear Function with 16 Segments

FIGURE CAPTIONS

Figure 1

(1) Computer in role of client on behalf of a user or user's process.

(2) Computer in role of server providing responses to requests from clients based on data from the data store (3)

(3) Data in custody of the server computer (2).

(4) Optional encryption of data in transit via an encryption key known to computers (1) and (2).

(5) Optional encryption of data at rest via an encryption key known to computer (2).

Figure 2

(1) Negotiation Agent Computer – Negotiation agent computers receive terms from the peers, assemble all the data, and then aggregate the peers' queries. The agent then infers responses that consider all the interests of the peers. These responses are shared with the peers who may then decide to proceed with a peer-peer transaction.

(2) Peer Computer – Peer computers can interact with each other on the network but require the service of the negotiation-agent computer to reach agreement before proceeding on peer-peer transactions. Each peer computer may have local data that is relevant in negotiation and queries or requests that it makes during negotiation. Together these constitute the peer's terms of negotiation. A peer will receive responses from the negotiation agent, which are the agent's inference from the terms provided by the peers.

(3) Peer-peer network transaction. Peers can use standard network protocols to conduct their business after making agreements through the negotiation agent.

(4) Optional concealment of negotiation terms and responses by means of encryption using an encryption key that is known by the peer computers but not by the negotiation agent.

Figure 3

(1) Partially encrypted documents containing concealed terms provided to the blind agent for negotiation. Each document represents the negotiation position of the party who has sent it to the blind agent. Encryption was performed using a session key that is denied to the blind agent. This key is the first session key of the improved process of Figure 4.

(2) Matching process step that compares encrypted terms from two parties to search for basis of agreement. The steps may be executed in sequence or in parallel as desired.
(3) Collation process that combines the results of the matching process steps.
(4) Combined basis for agreement document containing matching encrypted terms generated by all matching process steps.
(5) Distribution process that forwards an appropriate subset of the combined basis to each party where an appropriate subset contains those parts of the combined basis that include a concealed term offered by the recipient of the subset.
(6) An appropriate subset of the basis of agreement that will be sent to a party. This document is protected by the session key and cannot be read by the blind agent.

Figure 4
(1) Partially encrypted documents containing concealed terms provided to the blind agent for negotiation. Each document represents the negotiation position of the party who has sent it to the blind agent. Partial encryption was performed twice, first using a session key that is denied to the blind agent and second using a session key generated by and found only with the cloudwall (7).
(2) Matching process step that compares encrypted terms from two parties to search for basis of agreement. The steps may be executed in sequence or in parallel as desired.
(3) Collation process that combines the results of the matching process steps.
(4) Combined basis for agreement document containing matching encrypted terms generated by all matching process steps.
(5) Distribution process that forwards an appropriate subset of the combined basis to each party where an appropriate subset contains those parts of the combined basis that include a concealed term offered by the recipient of the subset.
(6) An appropriate subset of the basis of agreement intended for a party. This document is protected by two session keys and cannot be read by any party.
(7) The “cloudwall” component of the blind agent which is a small, highly-secure configuration of computer hardware. The cloudwall asserts a second partial encryption in step (10) on each document (9) using a second session key. Only the cloudwall has knowledge of the second session key.

(8) The “cloud” component of the blind agent consisting of a number of computers in an array or cloud and applied to the process of finding matching terms of agreement in the concealed negotiation positions.
(9) Partially encrypted documents containing concealed terms provided to the blind agent for negotiation. Each document represents the negotiation position of the party who has sent it to the blind agent. Encryption was performed using the first session key, which is denied to the blind agent. The document (9) in Figure 4 is identical to (1) in Figure 3.
(10) An encryption process that applies the partial encryption algorithm to (9) using the second session key generated and retained by (7).
(11) A decryption process that reverses the partial encryption step of (10) using the second session key.
(12) An appropriate subset of the basis of agreement that will be sent to a party. This document is protected by the first session key and cannot be read by the blind agent. The document (12) in Figure 4 is identical to (6) in Figure 3.

FIGURES

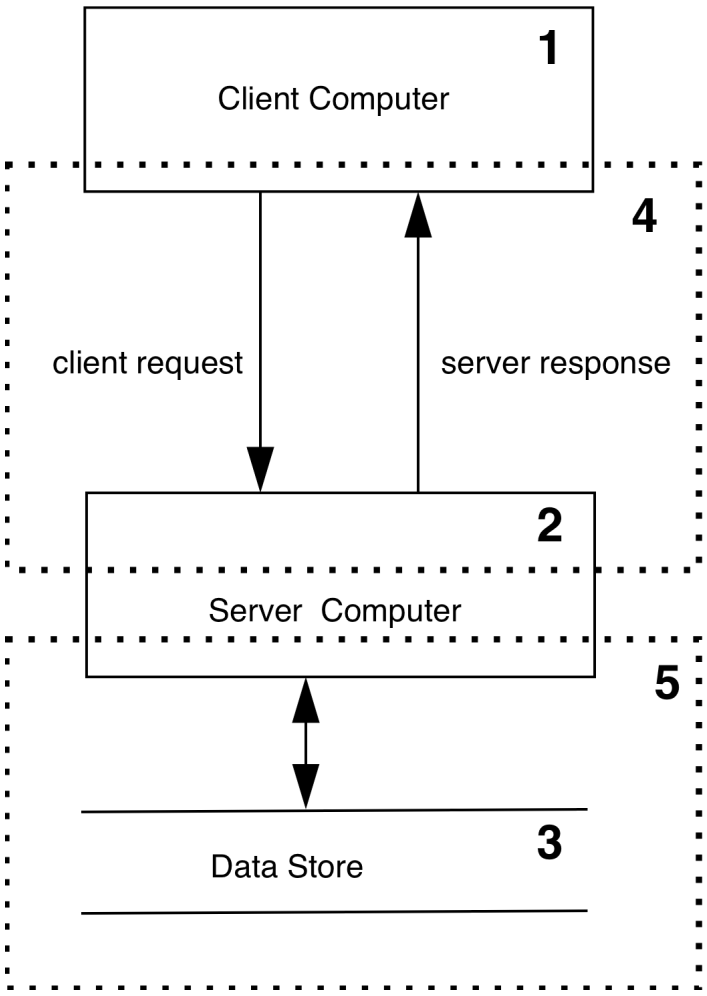
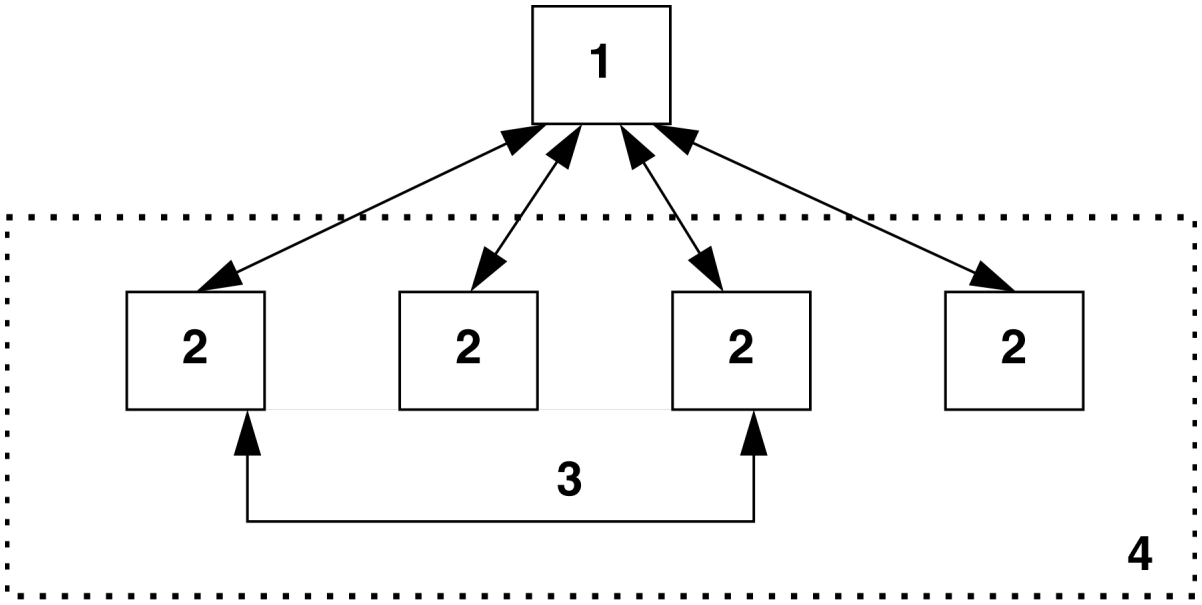
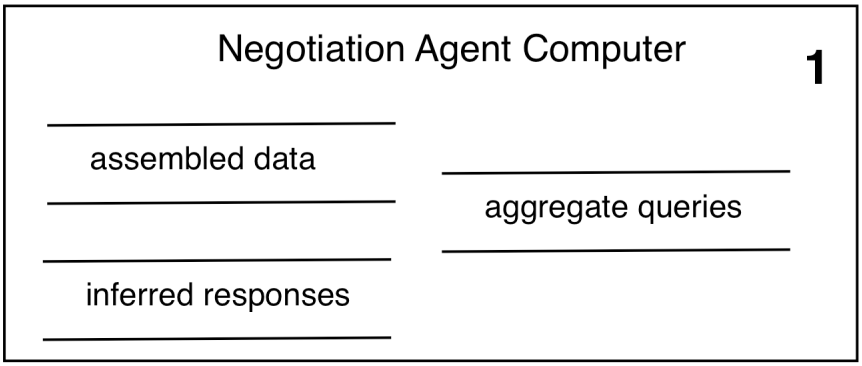


Figure 1


1
2
2
2
2
3
4


Negotiation Agent Computer
1
assembled data
aggregate queries
inferred responses

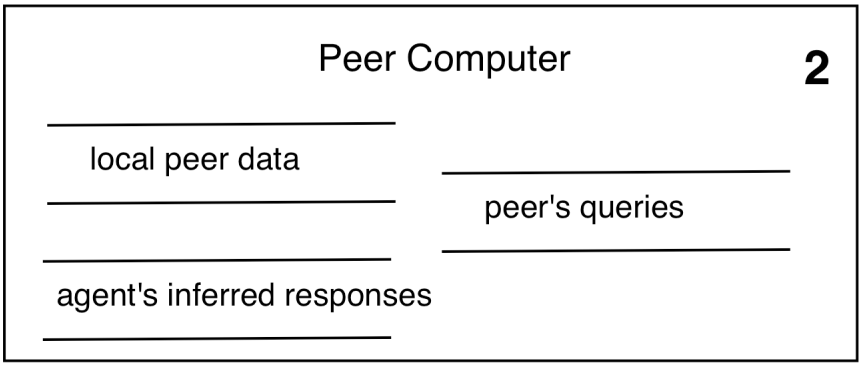

Peer Computer
2
local peer data
peer's queries
agent's inferred responses

Figure 2

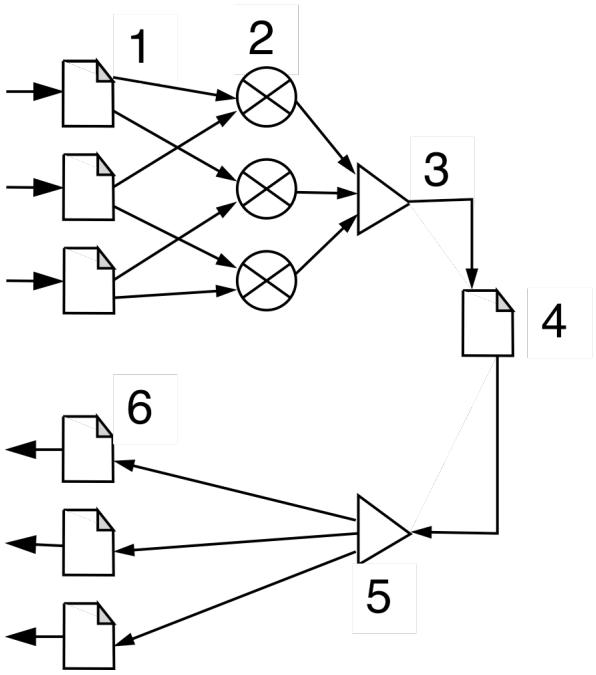


Figure 3

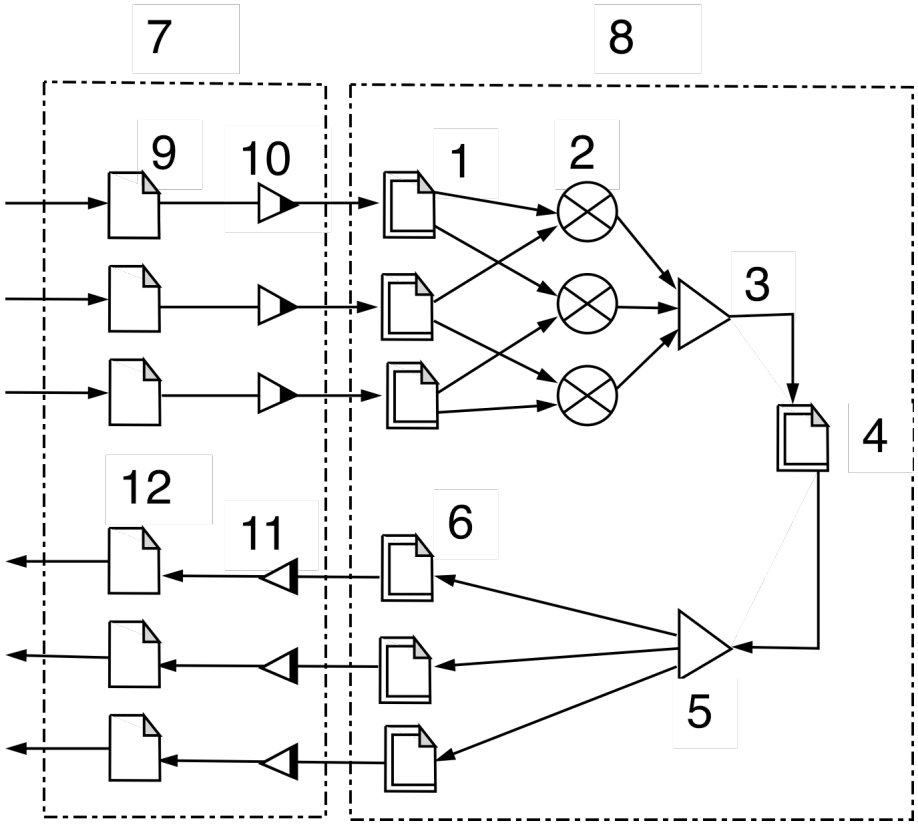


Figure 4

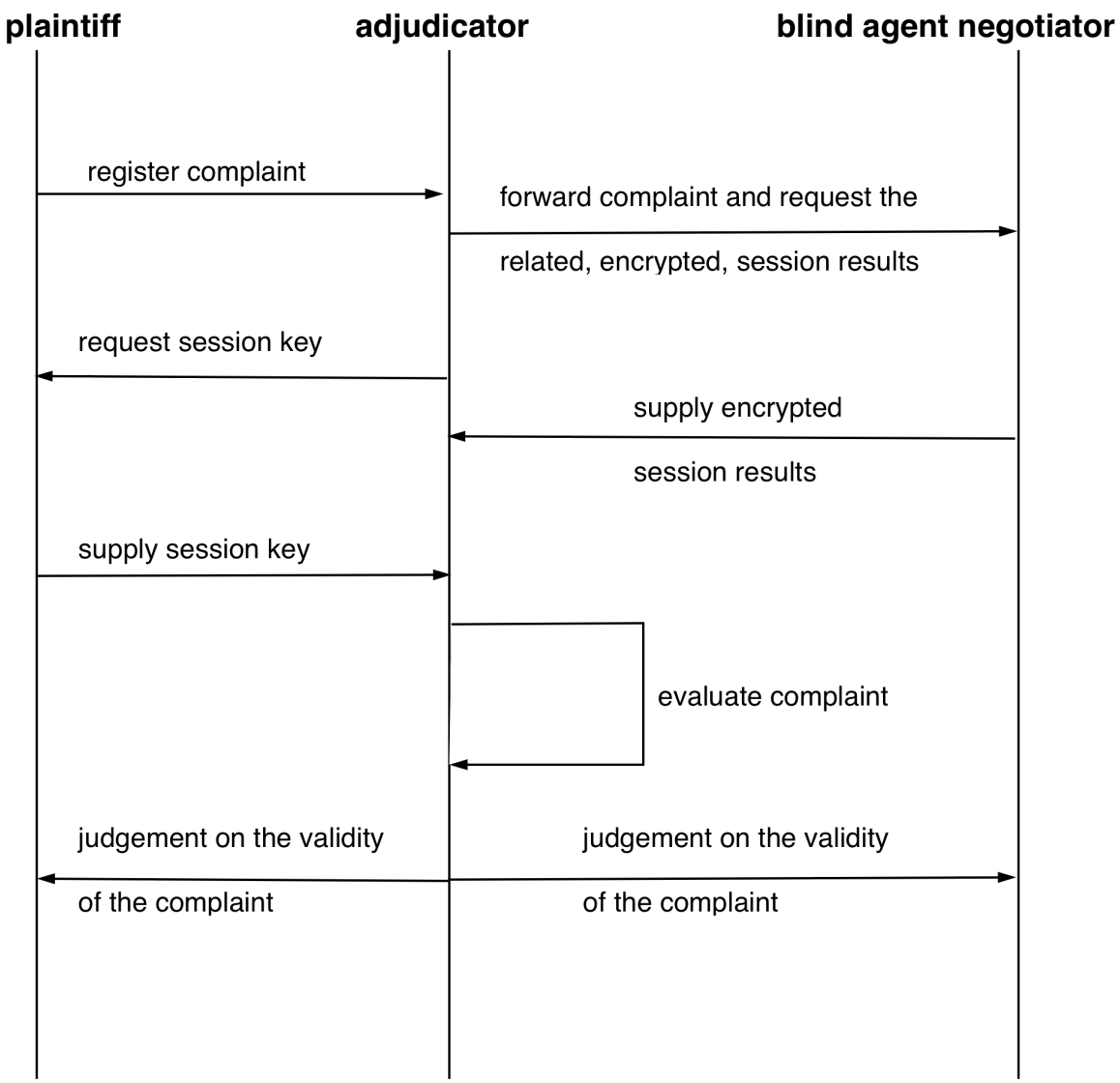

plaintiff
adjudicator
blind agent negotiator
register complaint
forward complaint and request the
related, encrypted, session results
request session key
supply encrypted
session results
supply session key
evaluate complaint
judgement on the validity
of the complaint
judgement on the validity
of the complaint

Figure 5

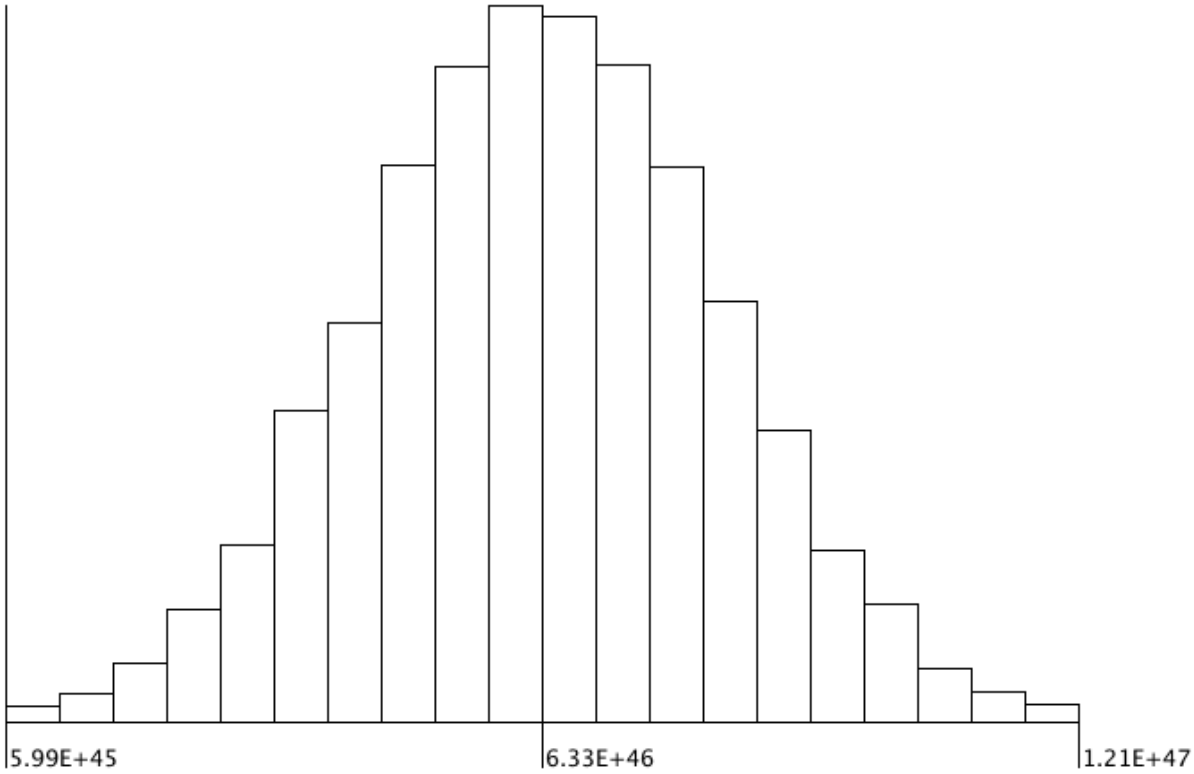


Figure 6

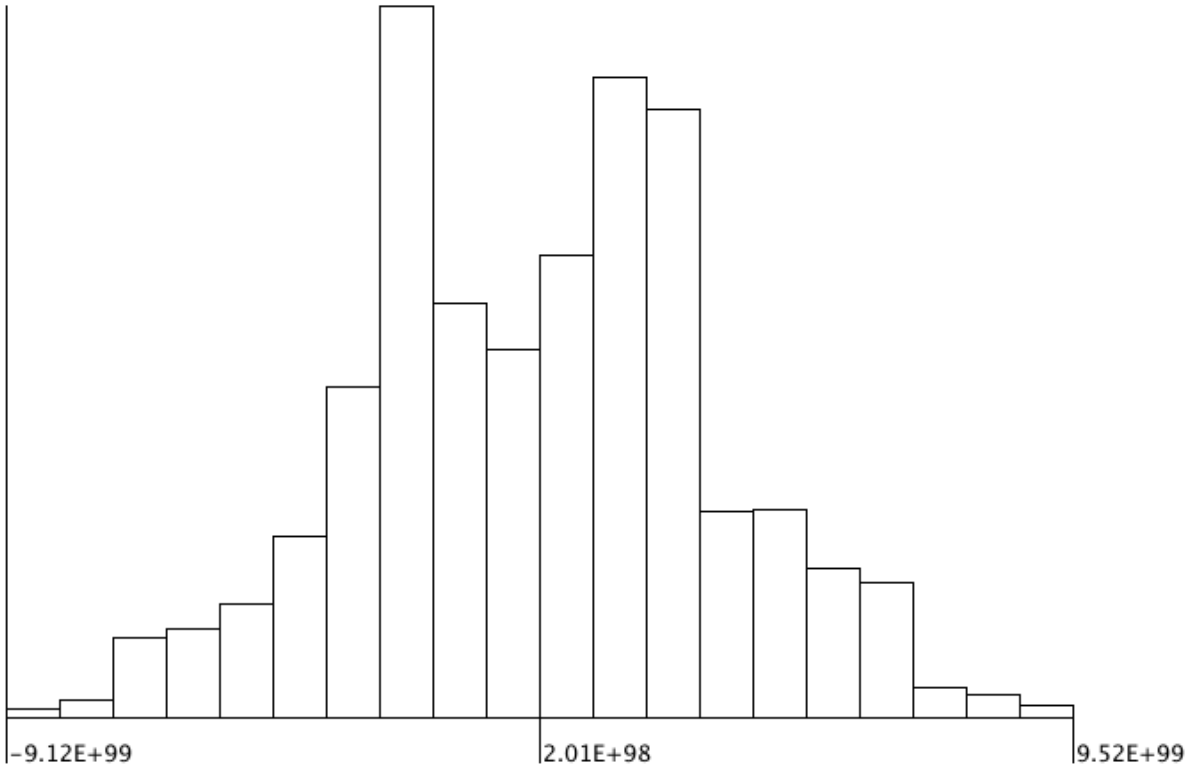


Figure 7