Anonymous, Authenticated Communication for Secure Sharing of SCADA and Control System Information

Timothy Draelos, Annie McIntyre, William Neumann, Richard Schroeppel
(tjdrael, amcinty, wneuman, rschroe)@sandia.gov
Sandia National Laboratories[1]

Christine Eliopoulos
celiopou@mitre.org
MITRE Corporation

Abstract: This paper describes the issues and solution options associated with providing anonymity to an authorized group of users who share digital information with a central collection/analysis/distribution center. Sharing of information within the SCADA and process control community is important for understanding threats and early detection of coordinated attacks, but anonymity and authentication of communication is crucial to enabling and maintaining trust in information sharing within the community. A complete solution addresses anonymization on three fronts: cryptographic message preparation, message communication, and message content. With our approach to cryptographic anonymity, the receiving center can ensure that an arriving message is from a member of the group, but neither the center nor an electronic eavesdropper can determine which member of the group sent the message. The sharing of sensitive information attributable to a particular user has risks that our cryptographic solution can mitigate. The network communication element of an anonymity solution incorporates the ability to anonymize the communication route of a message transmitted over the Internet. Since anonymous communication opens the door for untraceable system abuse, our approach to cryptographic anonymity employs a multi-level communication structure that mitigates abuse by allowing message revocation, yet retains true anonymity at the highest level. The use of operational procedures and/or filtering techniques to anonymize or sanitize message content is an equally important element to any solution. Anonymous, authenticated communication is an enabling technology to secure sharing of SCADA and process control system information.

Keywords: anonymity, authentication, information sharing, information security, process control, SCADA.

This work was supported under Award Number 2003-TK-TX-0003 from the U.S. Department of Homeland Security, Science and Technology Directorate. Points of view in this document are those of the authors and do not necessarily represent the official position of the U.S. Department of Homeland Security or the Science and Technology Directorate. The I3P is managed by Dartmouth College.

[1] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

1 Introduction

Cross-domain information sharing, in the context of a SCADA and Process Control community, is the exchange of information between two or more domains, where a domain is a collection of individuals, resources, and information owned by one entity (e.g., company) that requires protection (in this case anonymity) from other domains and eavesdroppers. Figure 1 depicts the various cross-domain environments within a SCADA and Process Control community.

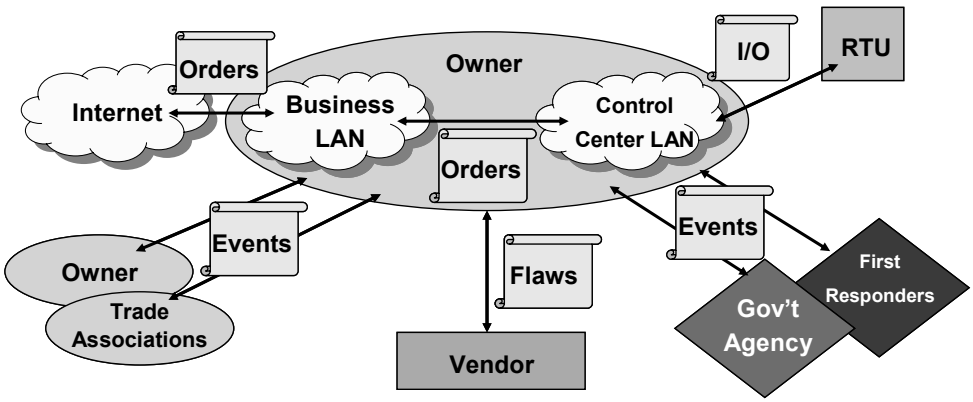

Figure 1 – Examples of information sharing across domains

Although information sharing can occur at various levels of communication and between various business sectors, this paper targets information sharing categories. Both of theses categories include on-going information sharing activities that are discussed in Sections 1.1 and 1.2 respectively.

- Information sharing between Process Control Systems (PCS) community members and government agencies.
- Information sharing among members of the PCS community.

Benefits to sharing information include identification of trends, realization of a coordinated incident, understanding potential technical effects and applying mitigations. Often, information sharing occurs between colleagues within a user community via telephone or in-person dialogue. However, large-scale, electronic sharing of information within a competitive industry is a different story, where information sharing can expose vulnerabilities at a specific site, erode public confidence and create a financial impact. In today's electronically networked age, information sharing must be viewed as a tool to protect assets and ensure uninterrupted operations and service.

The I3P[2] Task 1 Risk Characterization white paper [11] discusses threats and vulnerabilities to PCS systems, and their impacts on the overall business. Any information that can be used to prevent, recognize, and mitigate attacks on infrastructure systems proves valuable. The cost of inaction can be much greater in terms of downtime, public confidence levels, and equipment repair, than the costs of prevention and secure operations. The main objective of information sharing is to provide industry with the needed information to protect against, become aware of, and respond to threats. These threats can include overarching cyber, personnel, and physical security threats that can manifest as network attacks, software anomalies, and attempts at physical intrusion or compromise of information. Cross-domain information sharing provides a means to share information among industry asset owners and vendors that could help prevent, detect, or counter these threats. The types of information that could be shared with SCADA/Control System security in mind include system status (e.g., equipment failures), surveillance information, incident information (e.g., recent attacks, effects, and actions), and security solutions (e.g., secure configurations, best practices, etc.). This information can alert the industry to characteristics of a common incident and understanding of potential consequences during an event, as well as industry tested methods of responding to threats. The commonality and comparison of technical effects can be explored through data analysis.

An honest view of information sharing must acknowledge the risks as well as the benefits. An information sharing security requirement consistently communicated by industry stakeholders is anonymity of information providers [19]. The lack of an option for anonymous communication tends to suppress information sharing among competitors and industry participants. Requirements developed for a comprehensive information sharing system suggest incorporating anonymous, authenticated communication [19]. However, adding anonymity to an information sharing system must be done with care. Since anonymity can open the door for unattributable system abuse, revocable anonymity features should also be considered. Sharing information anonymously can mitigate the risk to industry while fostering the benefits. To achieve this anonymization, technology and procedures must be employed to ensure the protection of user identification at all steps during the sharing process. The discussion of anonymous, authenticated communication presented in this paper is applicable to the sharing of SCADA and control system information, and in general, to any other information sharing domain.

In Sections 1.1 and 1.2, we present several efforts to offer information sharing resources to the critical infrastructure community. Very little detail is publicly available about how these efforts address contributor anonymity. A summary of our understanding of the anonymity protection provided by one or more of the information sharing sites is as follows.

[2] I3P (Institute for Information Infrastructure Protection) is a national research consortium with a mission to identify and address critical research problems in information infrastructure protection (http://www.thei3p.org).

- The information sharing center, or hosting organization, vets and sanitizes information that is submitted. The identity of the submitting industry member is protected.
- The submitting industry member decides at the time of submission if the submitter is to be kept anonymous and/or the information forwarded onto a government or law enforcement agency. If the submitter desires anonymity, the information is sanitized so that no identifying description is included in the submission.
- Information is gathered from industry, aggregated, and anonymized by a third party security service. The resulting aggregate is then submitted to the information sharing center.

Anonymity is addressed at the time of submission and as the information sharing center processes the data. The methods discussed above do not address electronic identification protection while the data is in transit from the submitter to the center. Traceability to the submitter is technically feasible under these anonymity methods.

1.1 Information Sharing Between PCS Community Members and Government Agencies

Many asset owners within the PCS community feel that established information sharing solutions flowing from the owners to government agencies are sufficient. This information flow is regulated and procedures are in place to satisfy the reporting requirements. Some owners, however, would prefer more information flow from government agencies back to the owners. Government-hosted sites exist, or are being piloted, that provide general information to owners, including industry-wide alerts. Several of these sites are outlined below.

Homeland Security Information Network (HSIN)

This network, hosted by the Department of Homeland Security (DHS) National Operations Center [23], is a counter-terrorism communication network that connects all the US states and 50 major urban areas [22]. Situational awareness, information sharing with homeland security partners, analysis, and real-time capability, are some of the network's primary characteristics. The HSIN [24] for critical infrastructure, or HSIN-CI, is a pilot program, and includes information exchanged between DHS and private sector owners and operators. It also facilitates two-way information sharing with owners and operators, providing DHS with a base of locally knowledgeable experts and delivers real-time access to needed information while providing a pipeline to government. It is accessed by locally vetted membership, and sends alerts via wired and wireless phones, e-mail, fax, and pagers. Critical infrastructures participating include food and agriculture, water, energy, transportation, public health, banking, and telecommunications.

US-CERT®

The United States Computer Emergency Readiness Team (US-CERT) [25][26] is a partnership between the Department of Homeland Security and the public and private sectors. Established in 2003 to protect the nation's Internet infrastructure, US-CERT coordinates defense against and responses to cyber attacks across the nation. They offer alerts, tips, bulletins, and incident response services and serve as a center for response to major incidents, fostering of industry, government, and law enforcement collaboration, and hosting of protection working groups, vulnerability reporting capabilities, and training. Specifically, the Control Systems Security Center (CSSC) offers standards development, best practices, industry outreach, risk assessment methodologies, and vendor points of contact. The US-CERT web site provides the capability to report phishing, an incident, or vulnerability, and divides information by audience, both technical, non-technical, government, and control system users.

Infrastructure, Security, and Energy Restoration (ISER)

The ISER site is hosted by the Department of Energy (DOE) [21]. It contains news, emergency situation reports, lessons learned, weather watch capabilities, preparedness plans, and links to government offices, trade associations, and other industry organizations. This site encompasses information about natural disasters and emergency preparedness with a broader scope than data sharing for network attacks. Part of the Office of Electricity Delivery and Energy Reliability, ISER assists in preventing the exploitation of infrastructure by terrorists and provides resources for response to and recovery from disruptions.

Energy Information Sharing Analysis Center (ISAC)

The Energy ISAC [27] site provided security related information sharing for the oil and gas and other critical infrastructure sectors by interfacing with relevant agencies such as DHS and DOE through coordinating councils. The ISAC site was previously maintained by Science Applications International Corporation (SAIC), but it is no longer operational. Customers are directed to the World-Wide ISAC (WW/ISAC), which provides physical and cyber threat alerts from government agencies, law enforcement, and private security firms, to industry members.

1.2 Information Sharing Among PCS Community Members

Information sharing among member of the PCS community has been identified as a pressing need during discussions with key oil and gas stakeholders [19]. For example, shared data could provide base information for an organization to use in determining whether it is under attack. Likewise, if other organizations report issues, this may be an indication that a coordinated infrastructure attack is underway.

Control System Security Event Monitoring Working Group

The Control System Security Event Monitoring Working Group [14] is part of the Process Control Systems Forum (PCSF) and is chaired by Mr. Dale Peterson. The

working group utilizes data gathered and correlated by managed security services, and serves as a clearinghouse for information and tools that detect attacks on control systems. Their charter states that "The Working Group will (1) collect control system attacks statistics to quantify and qualify the threat (2) correlate control system detection events with IT detection events (3) normalize control system detection events from different vendors and (4) create and maintain a list of control system detection products and services". Threat statistics and trending is available on the PCSF website [15].

Industrial Security Incident Database (ISID)

British Columbia Institute of Technology maintains an industrial security incident database designed to track security-related incidents against process control networks and SCADA systems around the world [3]. This includes events such as accidental cyber-related impacts, as well as deliberate events such as external hacks, Denial of Service (DoS) attacks, and virus/worm infiltrations. Data is collected through research into publicly known incidents and from private reporting by member companies that wish to have access to the database.

1.3 Anonymous, Authenticated Information Sharing Issues

In this paper, we explore a range of anonymity issues associated with information sharing. We extend the original work of Beaver, Schroeppel, and Snyder [1], describing a method for electronically sharing information in an anonymous yet authenticated way and offer technical options for addressing anonymous communication and message content anonymity. The scenario we consider involves a group of users who wish to share information for common benefit, but the users wish to keep their identity unknown and independent from the data they contribute. They may also require that data received be only from an authorized user and that communications be kept private. One example is a hypothetical group of utility owners and vendors who wish to compile a database of experiences with computer incidents. This database might be situated in a server at a central location that has responsibility for group membership; data collection, analysis and distribution; and member and data security. The database would be a valuable resource to all, but a business owner with stockholder confidence to consider may share information only with the assurance of remaining anonymous. Furthermore, it would be important that only authorized users contribute information since a malicious user could submit bad information that could lead to adverse consequences. The users, therefore, do not trust other members of the group, much less the public (non-members), with their sensitive information, and they do not trust other members or the central location with attribution of their data. Members trust that communicated information will be kept confidential in transit and have integrity upon reception.

We consider the anonymity and authentication of the communicated message, the anonymity of the communication of the message over a public network, and the anonymity of the data itself. To create an anonymous message, one can simply type a letter and leave out identifying information. However, hints to the identity of the author may be found in the message content. Therefore, attention must first be paid to ensure that the content cannot identify the author. Sending an anonymous message can be as simple as sending the anonymous note in an envelope with no return address. An

electronic analog may be to send an unsigned e-mail message and strip the identifying information from the e-mail header. There are, however, further considerations. For example, a postmark can identify the general location from which the letter was sent and there must be an address for the recipient. In an electronic transmission, it may be possible to observe the network traffic. Even if the message itself has no identifying information, if someone could observe the path of the data, then there is no anonymity. Therefore, any system that hopes to keep its users truly anonymous must provide a method to ensure the anonymity of the author's identity as well as the anonymity of the electronic communication path.

In addition to anonymity, the issue of authentication is important in an information sharing scenario. If a system provides true anonymity, then there is also the opportunity to send false messages without attribution. In situations where action is taken on the basis of message content, false messages could cause undue panic and be very costly. Thus, messages must also be authenticated to ensure they come from an authorized user and that the message has not been corrupted since it left the source. This authentication to others can be done anonymously. One concern, however, is that even if messages are authenticated, there still may be system abuse by an authentic user. If messages can't be traced, then a malicious insider can't be detected.

These considerations lead to some design goals for authenticated, anonymous communication:

- anonymity of the message author,
- anonymous network communication paths,
- authentication of the source,
- anonymity and integrity of the data, and
- protection against system abuse by insiders through the revocation of erroneous messages.

The use of cryptographic anonymity as described in Beaver, Schroeppel, and Snyder [1] combined with prudent use of technology and procedures for network communications and message anonymity meets all of these goals and, in addition, provides for privacy during communication. An infrastructure center can ensure that each member of the information sharing group is endowed with the same information (i.e., keying material). Symmetry and therefore anonymity can then be maintained among the membership. Protection against system abuse by authorized members can be achieved by revocation of messages without revealing identities.

In Section 2, we introduce the elements of a complete anonymous, authenticated communication system and provide some background information and design motivation. The communication model for information sharing is presented in Section 3. The detailed technical discussions of the three system elements are presented in Sections 4-6.

- Section 4 – Cryptographic Protocols for Anonymous Authentication
- Section 5 – Anonymous Network Communication
- Section 6 – Anonymizing Message Content

Section 7 presents applications of our ideas and Section 8 offers important conclusions.

2 System Overview

In this section, we present the issues facing an anonymous, authenticated communication system and provide an overview of the three elements of a complete solution. We look at previous work to help us identify some important issues, and we examine the benefits and drawbacks of these schemes.

We identify three system elements that provide anonymity of information from its origin to its destination. First, using a set of protocols for anonymous authentication, the recipient of information at the destination can ensure that a received message was sent by an authorized member without gaining any knowledge of the identity of the sending member. Second, the communication path of the message between sender and receiver cannot be traceable if anonymity is to be maintained. Third, the content of the message, which will be revealed at the destination, must not expose the identity of the sender, either by the use of explicit identifying words or by the subtle use of keywords or phrases that distinguish the sender from other members.

These three elements of an anonymous, authenticated communication system (cryptographic message anonymity with authentication, network communication anonymity, and message content anonymity) are introduced separately below and treated in detail in Sections 4-6. In Section 4, we present a specific method of anonymous authentication that meets the design goals listed in Section 1.3. Sections 5 and 6 provide in-depth discussions of the issues and options for network communication and message content anonymity based on the needs and available resources of the user community.

2.1 Cryptographic Message Anonymity with Authentication

In an anonymous environment, the sender of a message is unidentifiable. If the users are supposed to be members of a distinguished group, then authentication is necessary to avoid use by non-members. It is also important to guarantee that a message has not been altered after it is sent by an authorized source.

One way to provide authentication is to have all the group members share a common piece of information. Showing knowledge of this information proves valid group membership. For example, users could share symmetric data authentication and encryption/decryption keys. When a member wished to send a message to another member, the first member would use the keys to encrypt and authenticate the message. The encrypted, authenticated message would be sent to another group member and if the authentication of the message verifies using the shared key, then the receiver would be sure that a valid group member had sent it. In addition, the sender of the message would be assured that only a valid group member would be able to decrypt the message.

This scheme preserves anonymity: as long as all group members use the same keys, no one knows which member encrypted and authenticated the message. Furthermore it provides data confidentiality and integrity.

Although this scheme is simple, there are drawbacks.

1. Group members must be trusted not to give their shared secrets away to non-group members. If this happened, neither the bad group member nor the unauthorized user who illegitimately obtained the keys could be identified.
2. The keys must be updated whenever a group member leaves or if the keys are suspected of being compromised.
3. The question of key distribution and update can be a serious problem if the group membership is very dynamic or if there is concern of key compromise.

There are several different types of authentication schemes that can preserve anonymity and address some of the issues raised above. For example, Schechter, Parnell, and Hartemink [17] use public key cryptography to construct verifiable common secret encoding to prove group membership. In their scheme, dynamic group membership is not an issue. One-time certificates or zero-knowledge proofs are other common methods (see [5] or [12]). However, these schemes are more complicated and/or involve more group interaction than our shared encryption/authentication key scheme. In general, zero knowledge protocols require at least three messages: offer, challenge, and proof. This is problematic in many information sharing environments, where the network might be under attack. Even worse are various distributed messaging methods (secret sharing), which usually require most of the members to be online during the protocol and may require communication of messages that are quadratic or cubic in the number of members.

As we will describe in Section 4.2, we use a simple method involving encryption and authentication using shared encryption and authentication keys, but instead of just having one pair of keys, we have a unique key generation protocol that makes key compromise difficult and key update easy.

2.1.1 Anonymity Revocation

In situations where true anonymity exists, the source of information is completely untraceable. This may cause undesirable situations. For example, troublemakers, insiders, and criminals can act without fear of identification. Instead of true anonymity, revocable anonymity may be preferable. In systems providing revocable anonymity, anonymity is in place unless a specified event (e.g., court order) demands it be revoked and the identity of the offender revealed. Key escrow is a common mechanism used to provide revocable anonymity. An interactive zero-knowledge scheme which supports identity escrow and key revocation (without having to issue new keys) is described in Boneh and Franklin in [2]. The interactive communication required is logarithmic in the number of users. The system also provides message unlinkability and allows users to be categorized into groups/subsets. Linkability is an undesirable characteristic in settings where traffic analysis is a threat.

2.2 Network Communication Anonymity

Even if the users within an information sharing group employ some method to authenticate their messages to each other anonymously, if others are able to view the network communications, it may be clear who the communicating parties were, thus rendering the anonymity controls ineffective. In order to preserve the anonymity provided by authentication mechanism, one must also employ some method for anonymizing the communications of the system. A number of such technologies have been proposed over the years. Chaum [4] provides one of the first discussions on the topic, introducing an anonymous e-mail forwarding system utilizing a public key encryption system. Since then, a number of systems have been proposed that offer different properties; Goldberg, Wagner, and Brewer [8] offer a survey of many of the currently available methods each of which offer a different set of anonymity and performance properties. The different performance properties are fairly simple to describe, and are usually quantified by the expected latency of a message passing through the system, and occasionally by the number of public or symmetric key encryptions that any machine in the system may be expected to perform. The anonymity properties, however, are a more complex matter. When describing anonymity, one must describe the type of anonymity being provided, and what kind of adversary may be trying to strip away the anonymity. A very good discussion of these issues can be found in Reiter and Rubin's work [16][14], but a summary is as follows.

There are three types of network communication anonymity that a system may provide: *sender anonymity*, where the identity of the sender of a message is hidden from some adversary; *receiver anonymity*, where the identity of the receiver of a message is hidden; and *sender/receiver unlinkability*, where even if both the sender and receiver are known to be performing some communications, they cannot be shown to be communicating with one another. The adversary in any of these scenarios may vary in strength and scope. For example, one may wish to protect sender anonymity against a simple receiver who only has access to the message and routing information that it received from its immediate neighbor in the communication network, while protecting sender/receiver unlinkability against an adversary who is able to monitor all messages, along with routing and timing information for some portion (perhaps the entirety) of the network.

In addition to these two notions, Reiter and Rubin [16][14] introduce the notion of degree of anonymity, which attempts to capture the level of assurance a communicating party should have that his identity has been exposed. The degree is described as an informal spectrum ranging from *absolute privacy*, where the adversary is unable to even detect that communication is taking place to *provably exposed*, where the adversary can not only determine the identity of the hidden party, but to also prove this identity to other parties. In between these two extremes, the authors describe three other degrees of anonymity: *beyond suspicion*, where the sender (respectively receiver or communicating pair) of a message is no more likely to be identified as the sender, than any other potential sender in the network; *probable innocence*, where the sender is no more likely to be identified as the sender than some other potential sender; and *possible innocence*, where from the adversary's perspective, there is some non-trivial probability that the true sender is some other party.

A few proposed anonymous communication systems are described in greater detail in Section 5 with regard to their basic operation and the level of protection they allow.

2.3 Message Content Anonymity

Ensuring that the content of a message does not reveal the identity of the source that created it is a non-cryptographic problem. When used in combination with crypto-based schemes for anonymous authentication and network communication anonymity, anonymizing message content provides an added level of privacy protection. A number of approaches can be taken to anonymize message content. These approaches range from simple procedures that a message originator can follow to sanitize content before it is released, to more sophisticated techniques involving message pre-processing and automated message sanitization. These techniques are discussed in greater detail in Section 6.

3 The Proposed Communication Model

There are several ways to authenticate anonymous communications. The cryptographic protocols used depend on the needs and attributes of the system model. A proposed design that is appropriate for the communication model described here involves protocols that are relatively simple, secure, flexible, and that minimize key update problems if the group membership is dynamic. Furthermore, group interaction is not required. In this section, we describe the model and in Section 4, we describe the cryptographic protocols.

In our model, we assume that a specified group of mutually distrustful users wish to communicate. The users may communicate by posting information on some common electronic bulletin board, by sending each other e-mail or by building a shared database to collect the data. In any case, the users send the information electronically to some central location or Center which either stores the information or redistributes it. We first assume that this central location is trusted and later explain how to avoid this assumption if there is no point of trust in the system.

It may be that individual users who communicate among themselves wish to have this authenticated, anonymous communication capability. We broaden our view beyond individual users to allow for each "user" to be an entire group (e.g., a business). A group may have multiple users at a site, yet wish to be identified as a single entity. We define a collection of users identified as a single entity to be a *domain*. A domain may also be a single user. We define *domain-to-domain communication* to be the communication between domains. The protocols for cryptographic anonymity described in Section 4 provide for the communication of strictly anonymous, authenticated messages between the domains, assuming the underlying network provides sender anonymity for messages sent from a domain to the Center, and receiver anonymity for replies from the Center to a domain. The adversary in this model will vary for real instantiations from any single domain up to any coalition of up to n-2 domains, where n is the total number of domains in the system.

If a domain consists of multiple users (e.g., a company with several employees using the software), then we recognize that communications that must be anonymous to other domains may not need to be anonymous to others within the same domain; certainly a business expects control over the communications of its employees. Hence, besides describing domain-to-domain communication, we also define *intra-domain* communication. Intra-domain communications can be authenticated, and can provide anonymity, revocable anonymity or no anonymity. This two-level structure is what enables the ability to revoke messages anonymously at the domain-to-domain communication level. A high level description of the model is given in Figure 2, and a description of the protocols that support inter-domain and intra-domain communication can be found in Section 4. A description of how the central authority may be eliminated is covered in Section 4.3.

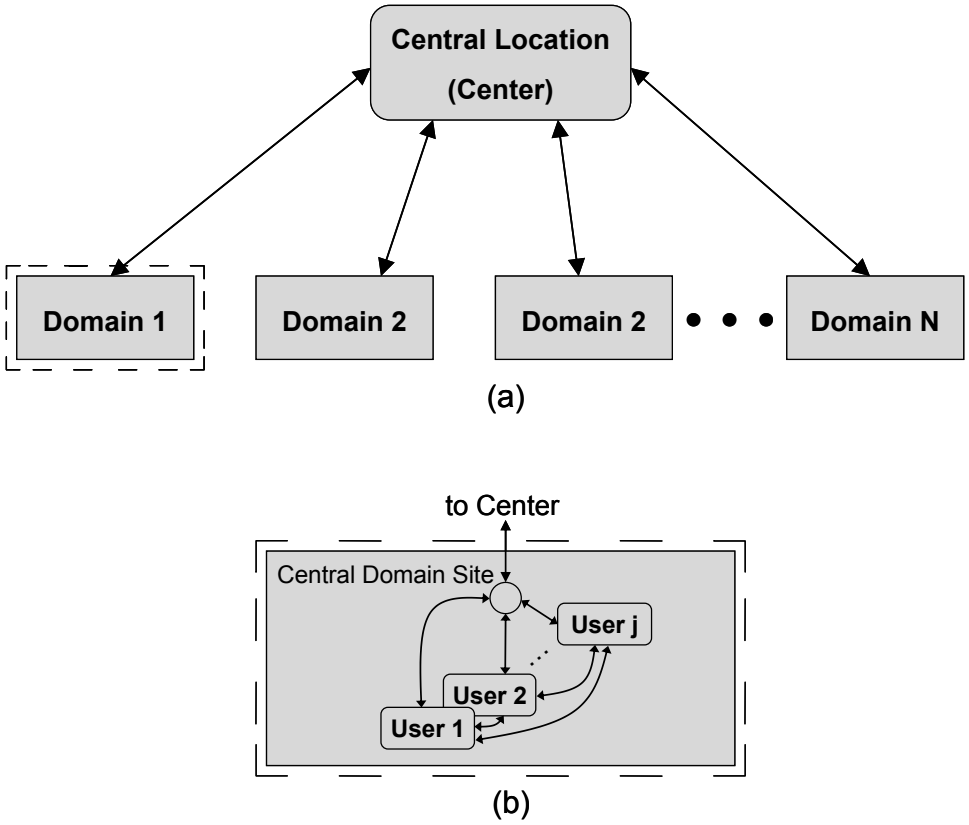

Figure 2 – High level anonymous, authenticated communication model: (a) Inter-Domain communication, (b) Intra-Domain communication.

3.1 Intra-Domain Communication

Within each domain, there may be multiple users and the need for anonymity may not be as rigorous within the domain as it is between domains. For example, an employee usually does not expect to have strictly anonymous communication and in fact the company usually prefers to monitor their workers. Hence, communication within the domain can be more flexible: they may be open (authenticated, but not anonymous); anonymous with revocable anonymity features (where the identity of the malicious user can be learned); or strictly anonymous as described above. Each domain can select different communication rules for use within their domain.

3.1.1 Domain Committees

We assume that we can form a trusted committee from among the users in a domain to perform the role of message revocation. This is natural in a setting where the domain consists of a group of users belonging to the same business. For example, the committee could consist of the Vice Presidents or some trusted members who serve as a review and approval board, etc.

3.1.2 Anonymous Message Revocation

With the Domain Committees in place, we are ready to describe the anonymous message revocation protocol. It is enabled via a "revocation token" that the Center sends back to the domain committee when a message is received.

How the domain committee decides which messages to revoke and what the Center does with the revoked message is a policy matter to be determined by the users beforehand. For example, the message could be revoked simply upon majority vote by the domain committee. When the Center learns that a message has been revoked, it could simply flag the message or it could remove it from the database and/or send notification of the revocation to all domains. The practice of having a committee review an important document for approval is not uncommon and so we believe this procedure is rather natural in the settings we envision. It is important to note that at no time does the Center learn the identity of the domain that originated the message. Within the domain, the level of anonymity can vary. We describe three options next.

3.1.3 Intra-domain communication with strict anonymity

If the users wish to maintain anonymity within their domain, the protocols described briefly in Section 2.1 and in detail in Section 4 are run without modification. If a domain committee decides a message is bad (e.g., not credible), they can revoke it using the revocation token, but they have no way to learn the identity of the sender within their domain. If this is not desirable, there are alternatives as described below.

3.1.4 Intra-domain communication without anonymity

In this scenario the users will not have anonymity between users within their domain. When the user sends a message, he follows the steps in Section 4.2. However, before

the message is sent the domain's anonymous communication infrastructure logs a copy of the message along with the identity of the sender. At any time any user in the domain can check and see who sent which messages. In particular, when the domain committee receives notice that a message has been sent from their domain, they can check the logs of their network, find out who sent the message, and take appropriate action if the message turns out to be bad.

3.1.5 Intra-domain communication with revocable anonymity

In this scenario the users will have anonymity unless there is concrete suspicion of abuse. We assume that the domain is equipped with a public-private key pair, where the private key is split among the domain-authorized committee members. Alternatively, the domain-committee may employ a threshold scheme to share the key (see [13]). This means that something that has been encrypted with the domain public key can only be decrypted if all of the committee members get together to reconstruct the private key. As before a user sends a message following the protocols in Section 4.2. In this case, the message header (before it is sent) includes an identifier of the machine that sent the original message. This identifier is randomly padded and encrypted with the public key of the domain and appended to the message. Note that the anonymity of the message is still retained because only the sending domain has the corresponding private key that could decrypt the identifier. The message and identifiers are not logged. All identifiers are then removed and the message is sent. If, upon receipt of the return reply message and revocation token, the domain committee members agree the message is invalid, the private key of the domain may be reconstructed and used to decrypt the identifier that was appended to the message. They can then identify the domain user who sent the false message and appropriate action can be taken.

4 Cryptographic Protocols for Anonymous Authentication

Cryptography can be used to offer a technical solution to anonymous, authenticated communication assuming the sender's location cannot be traced and the message contents do not give away the sender's identify. We use a simple method involving encryption and authentication using shared encryption and authentication keys to ensure anonymity of authorized members. However, instead of just having one pair of keys, we have a unique key generation protocol that makes key compromise difficult and key update easy.

4.1 PKI Assumptions

We assume that some form of a Public Key Infrastructure (PKI) is in place (a full description of a PKI is out of scope for this paper). In particular, each domain is equipped with a public/private key pair. Since any domain has access to another domain's public key, messages are allowed to be sent to any specific domain securely. In what follows we will mention use of hash algorithms and encryption algorithms. Although the choice of algorithms is independent of our protocols for cryptographic

anonymity, we recommend using a hash algorithm such as SHA-256 and the advanced encryption standard, AES, with 128-bit keys for encryption.

4.2 Anonymously Authenticating the Message

We chose to implement a form of authentication that uses symmetric keys for encryption/decryption authentication. As mentioned above, one drawback to this is the problem of using one key. Instead of a single encryption key or even a set of encryption keys, we developed a protocol for generating a valid encryption key and authentication key. In order to do so, the user must have a physical object (such as a CD, DVD, or some portable read-only memory device) as well as special knowledge of a token that changes frequently. When a message is sent, it is encrypted with a key generated in this way and the message is accepted as valid if it can be decrypted properly. Again, for now we assume that the system has a trusted Center, but we provide protection against a Center that tries to break contributor anonymity.

4.2.1 Protocol: Setup and Update

Output: Random Number CD.

1. The Center generates a Compact Disk (CD) with (at least) 20,000 randomly chosen 256-bit numbers.
2. Each legitimate user is given a copy of the Random Number CD.
3. The hash value of the contents of the CD are published in some mutually agreed upon location, for example in a newspaper or a public web site.
4. Each user computes the hash value of the contents of their CD and compares it to the published hash value.
5. The user accepts the CD if and only if the hash values match.

4.2.2 Protocol: Daily Token Generation and Verification

Output: Daily Token.

1. Once daily the trusted Center generates a "Center token", T. A Center token is a randomly generated number, much like a password for the day. T is a randomly chosen 128-bit number.
2. The daily Center token, T, is encrypted using the public keys of each domain and sent to each domain:
   (a) Let the public key for domain i be denoted K_i and denote by $E(K_i, m)$ the message m encrypted with public key algorithm E and key K_i. For example, E = RSA encryption.
   (b) Form $M_i = E(K_i, T)$ for each domain i.
   (c) Send the message $(M_1, \ldots, M_n)$ to each domain.
   (d) Publish the value of the hash of $(M_1, \ldots, M_n)$ in some public location.

3. Upon receipt of the daily Center token, T, each domain checks that it received the same token as every other domain:
   (a) Each domain i decrypts M_i using their public key. Call the result T'.
   (b) Each domain checks that the hash of $(M_1, \ldots, M_n)$ matches the published hash value. (Step 2d above)
   (c) For all $j \neq i$, domain i checks to see if $E(K_j, T') = M_j$. If this checks for all j, then domain i accepts T' as the daily token T.

The Random Number CD and the daily token are the input into the key generation protocol. In order to ensure anonymity, all domains must be certain that they are all using the same inputs. Verification that all domains have the same CD is carried out in Steps 3, 4, and 5 of protocol Setup and Update. Since a hash algorithm is essentially a one-way function, the hash of the CD can be published in some public location. The hash value will not reveal any information about the content of the CD. Hence the information on the CD remains secret, yet the users can confidently verify they all have the same CD. The proof that each domain receives the same daily token from the Center is carried out in Step 3 of the Daily token generation and verification protocol.

The writing of the hash of the CD and the hash of the Daily Token to a public place must include the time of the writing and enforce append-only operations to prevent the Center from distributing different CDs/Daily Tokens to different domains. If the Center is able to distribute different key material to different users, it could defeat the expected anonymity process and identify senders of messages. This model assumes that the Center is a trusted authority and in particular can be trusted to generate the Center token. If that is not the case, then the group of users can generate the token in a distributed manner as referenced in Section 4.3.

4.2.3 Protocol: Key Generation, Message Encryption and Authentication

Input: Message, Date, Daily Token, Random Number CD.
Output: Cryptographically protected message file, CP-MSG.

1. Generate a data encryption key and data authentication key:
   (a) Insert the Random Number CD.
   (b) DONE=FALSE
   (c) Repeat until DONE
      i) Generate a random number, r between 1 and N (where there are N random numbers on the CD).
      ii) Read in the r-th random number on the CD, n_r.
      iii) Compute $h_1 = Hash(n_r)$ and $h_2 = Hash(T)$ (where T is the daily token).
      iv) If the last ten bits of h_1 equal the last ten bits of h_2, then DONE=TRUE.
   (d) The encryption key for this message is $K_E = \text{Hash}(T \,||\, n_r \,||\, T)$.

(e) The authentication key for this message is computed by encrypting the encryption key, K_E, with itself, $K_A = E(K_E, K_E)$. Here E is a symmetric key encryption algorithm such as AES.

2. Encrypt the message: $M = E(K_E, message)$.
3. Create a message authentication code (MAC) of the encrypted message: $A = MAC(K_A, M)$. Here *MAC* is a symmetric key data authentication algorithm such as HMAC.
4. If anonymous message revocation is being used (see Section 3.1.2), and the router doesn't handle this, create a random reply key *RK*, encrypt it $E(K_E, RK)$, and concatenate it to the encrypted message prior to generating the MAC.
5. Create a cryptographically protected file, CP-MSG, containing the encrypted message, *M,* the index of the random number used in Step 1c, *r*, the date, the encrypted reply key, *RK* (if applicable), and the MAC of the encrypted message, *A*, or $MAC(K_A, A \,||\, E(K_E, RK))$.

The file CP-MSG must contain the index of the random number used in Step 1c and the date so the recipient can reconstruct the key used for encryption in order to decrypt. When the message is sent through the anonymous communication channel (described in Section 5), a one-time random key is appended to the message and then the entire message is encrypted with the public key of the recipient. It is expected that the file CP-MSG will be encrypted with the Center's public key prior to transmission.

Requiring both the possession of the CD and the knowledge of a daily token sent only to the valid users minimizes the risk of key compromise. A stolen CD is no good to thieves unless they also have access to the daily token. In addition, the contents of the CD could be encrypted to prevent unauthorized access to its key material.

Updating the token daily is just a suggestion; the token can be updated as frequently (or infrequently) as desired to ensure that only valid users have access to the correct Center token. To mitigate the possibility of someone stealing a CD, there can be policies in place to require physical control of the CD. This may be acceptable if these incidents are not too frequent and would help detect a stolen CD which would alert users to a potential problem.

Note that it is not possible for someone to generate a valid key with just knowledge of a few of the keys on the CD: if someone copies down some of the random numbers off the CD (to avoid detection of a stolen CD), the odds are extremely unlikely that those numbers will be valid numbers for the day's token (Step 1c of the Key Generation, Message Encryption and Authentication protocol). Thus, even if a person knows some of the keys and the token, the probability is very small that he or she would be able to generate a valid encryption key. We should note however that the condition required in Step 1c of the Key Generation, Message Encryption and Authentication protocol is not too restrictive for someone in possession of the CD: if the CD contains 20,000 values, then on average 20 of the numbers on the CD will pass the test in Step 1c. These strategies minimize the need to update the CD when users leave and so minimizes key management problems if the group membership is changing.

Finally we note that the message encryption also achieves the design goal of ensuring data integrity (a corrupted message will not decrypt) and provides for privacy during communication.

4.2.4 Protocol: Message Decryption and Authentication

Input: Cryptographically protected message file, CP-MSG, Random number CD.
Output: Either decrypted message or invalid message warning.

1. Upon receipt of the file, CP-MSG, read in the date, the index, r, the encrypted message M, the encrypted reply key (if applicable), and the MAC, A.
2. Let T be the daily token corresponding to the date.
3. Let n be the r-th random number from the CD.
4. Let $h_1 = Hash(n)$ and $h_2 = Hash(T)$
5. If the last ten bits of h_1 DO NOT equal the last ten bits of h_2, then reject the message as invalid.
6. Otherwise, Let $K_E = Hash(T \,||\, n_r \,||\, T)$ and $K_A = E(K_E, K_E)$.
7. Validate the MAC. If $MAC(K_A, M) = A$, decrypt the message $D(K_E, M)$.
8. If anonymous message revocation is being used, see Section 4.2.5 about additional message processing

Revocable anonymity mitigates abuses by insiders in the system, but at a cost of confidence in true anonymity. The decision of whether or not to revoke anonymity depends on the situation. For example, within a company, employees may desire anonymity when communicating electronically, but the company may have the right to revoke anonymity if wrongdoing is suspected. On the other hand, if a community of competitive companies is contributing information to a joint venture, they may want their communications to be strictly anonymous with no possibility of revocation in order to protect proprietary information. Hence, instead of revocation of anonymity we provide a protocol that enables anonymous message revocation. A bad message can be revoked, but there is no chance that the identity of the sender can be revealed.

Anonymous message revocation for a particular domain is the responsibility of a Domain Committee made up stakeholders for that domain (see Section 3.1.1). The revocation protocol is enabled via a "revocation token" that the Center sends back to the domain committee when a message is received. This is possible since an anonymous return reply envelope and a reply key are included in the message.

4.2.5 Protocol: Anonymous Message Revocation

1. When the Center receives a message, it will have been encrypted with its public key. The Center decrypts the message and learns the content of the file CP-MSG (see Step 3 of the Key Generation and Encryption protocol) as well as a one-time random 128-bit encryption reply key, *RK*.
2. The Center runs the Decryption and Authentication protocol on the CP-MSG file. If the authentication is not successful, the message is rejected and the protocol terminates. Otherwise, the message is decrypted.
3. The Center generates a random 128-bit revocation token.
4. The Center encrypts the token and a copy of the message using the reply key, *RK,* and sends it back to the message originator in the return reply envelope that came with the message. Note that the network communication path is protected against traceability to avoid identifying the sender of the original message.
5. When a domain receives the reply, it sends a copy of the message to every member of the domain committee.
6. The committee retrieves a copy of the reply key, *RK,* from the domain onion-router, checks the authentication of the message, and decrypts the message if valid.
7. The domain committee decides if the message is legitimate. If so, it does nothing, if not, it sends the revocation token back to the Center.
8. If the revocation token is sent back, the Center marks the message as revoked.

4.3 Non-Trusted Authority Model

The trusted central authority (Center) in our model is needed only for secure token generation and distribution. If there is no trust in the system, the token generation can be done by the users themselves, using a distributed, threshold secret sharing method (see, for example, [13]). In a threshold secret sharing scheme, the users can even tolerate a number of malicious players up to some threshold; hence this will work if the users don't trust each other or if some of the users are unavailable during the protocol.

Finally, if the token generation is done this way, communication is between domains only, thus eliminating the need for the Center. In this case, the encrypted, authenticated message is sent to a particular domain instead of the Center and the message can be decrypted and its authentication verified right at the domain. The receiving domain then carries out the Anonymous Message Revocation protocol and sends out the revocation token. The receiving domain knows to revoke the message if the revocation token is returned.

4.3.1 Initialization and System Update

1. When the system is first initialized, a compromise is detected, or a system refresh is desired, the Setup and Update protocol should be carried out.
2. Once daily, the protocol Daily Token Generation and Verification should be carried out.

4.3.2 Sending a Message

1. *M* is processed to eliminate clues of the author as described in Section 6.
2. *M* is encrypted using the Key Generation, Message Encryption and Authentication protocol, which produces the file CP-MSG.
3. The file CP-MSG together with the return reply envelope and the reply key, *RK,* is sent via the onion-router to the Center. The domain prepares and logs the outgoing message depending on the type of communication desired in the domain.
4. Upon receipt at the Center, the Decryption and Authentication protocol is carried out. If the message is determined to be invalid, it is rejected and the process stops.
5. The Center carries out the protocol Anonymous Message Revocation.
6. If the domain committees revoked the message, they may decide to investigate the identity of the user within their domain depending on their chosen communication scheme.

4.3.3 Retrieving Information

The protocols described above outline a method for sending information to a common location; however, how the information is disseminated is another issue. We assume that all valid users have access to the information, which allows the Center to redistribute the information to each domain encrypted with the domain's public key. Alternatively, the users can follow the above procedures, sending the messages to each other instead of the Center.

5 Anonymous Network Communication

Assuming the presence of a central authority (Center), all communications will be sent electronically to this central site. Even though all identifying information has been removed from the message, an observer of the network may trace the path of the message from the sending domain to the Center, effectively removing the anonymity. In Section 2.2, we introduced the need to anonymize network communication between senders or between the Center and a sender. We now will describe a few of the anonymous network communication techniques that have been developed over the years and elaborate on the protections they offer.

Note that any of these anonymous channels can be used with the anonymous authentication system described in Section 4. The systems described here provide

varying levels of anonymity and flexibility and the choice of a communication channel used to transmit messages should be made, based on the properties one requires of the system. While the choice can vary depending on expected level of traffic analysis by adversaries, we require that whichever method is chosen provide an anonymous return reply capability, though this restriction may be loosened in certain situations, which we describe where appropriate.

The simplest mechanism is called a Type 0 remailer, a system designed to allow a user to send an e-mail where the recipient is unable to determine the identity of the sender. This remailer simply strips identifying headers off the e-mail and forwards the message to the intended recipient. The originator's IP address is not revealed to the recipient as it is sent through an intermediate server, providing sender anonymity against a simple adversary. However, this does not protect against someone who can observe the communications into and out of the remailer server, nor does it protect against malicious servers.

Building on the Type 0 remailer is a Type 1 remailer, a system similar to Chaum's [4]. Here the e-mail message consists of a nested set of encrypted messages and is sent through a path of specialized re-encrypting routers called mixes. In order to make traffic analysis more difficult, the mixes forward a message only after receiving N messages. Type 1 remailers provide sender anonymity, improved traffic analysis resistance, and security against malicious mix servers. However, they are still subject to spam attacks, more advanced traffic analysis, and replay attacks.

A Type 2 remailer, sometimes called a mixmaster, is similar to a Type 1, but offers sender/receiver unlinkabiity and improves resistance to traffic analysis and replay attacks by using message padding, delay, and reordering. There are several variants on these types of systems, some allowing the sender to include an anonymous return envelope so the receiver can respond to the anonymous e-mail without knowing the sender's address. The Cypherpunks have a mixmaster remailer network with about twenty nodes, processing about 6500 messages per day, where forwarding latency per hop ranges from a few minutes to hours. The software is available for free download, including source code from the cypherpunks website [20]. Also available are links to a number of other remailer sites, although at the time of this writing, only one of those, Dizum, remains in operation.

One of the drawbacks with these systems is that they were designed solely to handle e-mail messages, the primary form of Internet communication at the time they were developed and are not particularly efficient when two-way communication is desired.

As other forms of Internet communication, particularly http traffic, became more prevalent on the Internet, more general methods of anonymous network communication were proposed. One of these was the Crowds project from AT&T [16] that created loose associations of users who would randomly forward messages amongst themselves, creating a random path from sender to receiver. This technique offers a form of sender anonymity in that any request is equally likely to have come from any member of the crowd. It does not provide for traffic analysis resistance from an adversary with a global view of the communications, and it is vulnerable to limited application of timing analysis.

Perhaps the most widespread anonymous network communication technology today is the onion router [18], [9]. Onion routing is an architecture made up of proxy servers that limits a network's vulnerability to traffic analysis while providing near real-time, bi-directional, anonymous communication for any protocol that can be adapted to use a proxy service (e.g., e-mail, ftp, http, etc.). In this architecture, no one but the initiator's proxy server knows anything but the previous and next hops in the communication chain. This implies that neither the respondent nor his proxy server, nor any external observer need know the identity of the initiator or his proxy server. This is done by padding and encrypting not just message content, but routing information as well. When done properly, each server along the route learns only which server was before it in the routing chain and which server comes next. Thus, if a single properly behaving server exists in the route from sender to receiver, then sender anonymity is preserved. Additionally, each message is re-encrypted at each hop and forwarded after some random delay. Dummy packets can also be used during communication in an effort to thwart traffic analysis. A modern onion router implementation, Tor [6], provided by the Electronic Frontier Foundation, incorporates hundreds of proxy-servers, and is used by hundreds of thousands of people worldwide.

When making a choice of anonymous communication infrastructure, one should consider how much delay in delivery is acceptable and the plausibility of various attacks given likely adversaries. For the proposed communication model described in Section 3, we suggest the use of an onion router to process the messages. One reason for suggesting this is the anonymous reply capability: when a message is sent, a "reply onion" can be included that allows the receiver to send a reply back to the original sender while preserving the anonymity of both parties. We assume that each domain has an onion router on site through which all of its messages pass initially. The onion router may or may not log the message (see Section 3.1), but we do assume that when a message passes through the domain onion router, it appends a randomly generated single-use reply key, *RK,* to the message, logs a copy of *RK* at the domain, and then encrypts the message and *RK* with the public key of the intended receiver. This ensures that only the recipient can read the message and also will aid in the message revocation procedure described in Section 4.2.5.

Note that the requirement for the communication infrastructure to support anonymous replies can be removed if the Center posts the reply message (encrypted if privacy is required), along with a revocation token that is encrypted with the domain-supplied reply key, *RK*, to some bulletin board-like site or RSS-style syndication point that may be read by all of the domains. So long as the access logs of this bulletin board are protected from an adversary, anonymity will be preserved while allowing for message revocation.

6 Anonymizing Message Content

The use of cryptographic protocols alone does not guarantee that information cannot be traced to its source. Neither do the techniques for obscuring the details of network communication fully protect the identity of a message sender. Techniques for anonymizing message content will afford information providers an added measure of

privacy not offered via the crypto-based anonymity techniques and the anonymous network communication techniques described above.

These non-crypto-based techniques include establishing clear policy and operational procedures for information release and training users on how to implement policy guidance when releasing products outside the local domain. Additionally, pre-processing techniques can be used to detect and prevent attempts (inadvertent or malicious) to release sensitive information that could result in compromised anonymity.

6.1 Policy, Operational Procedures, and Training

Central to an organization's ability to anonymize information prior to external release is its ability to articulate clear policy governing information sensitivity and sanitization rules. Without policy to guide users as they prepare products for release, decisions about what information to include and what to leave out or obscure are left to the discretion of individual users. The application of inconsistent guidance when releasing information will result in poor or degraded data quality at a minimum. In the absence of clear policy guidance, anonymity compromises are inevitable. A flexible policy may also be needed to accommodate changes in sharing agreements.

Policy must be specific to the type(s) of information being released and the circumstances under which it is being released. An example of this might involve the release of cyber incident data outside the local domain for purposes of correlation and analysis. The organization releasing the information must release enough information about the incident to facilitate meaningful analyses while also preserving its anonymity. Company policy governing the release of details about an incident should identify what and how information should be omitted or obscured in the incident report. Details from the report can be removed altogether or "fuzzed" to hide the identity of the source. Information that could identify persons or places or link events to specific individuals or groups should be omitted when anonymity is desired. Often, dates and times can be enough to link information with a particular user or domain. Under certain circumstances, however, complete disclosure of all details of an incident may be necessary to enable critical emergency response or to assist analysts in halting the spread of an infrastructure-wide attack in progress.

Policy intended to preserve anonymity should also prohibit the use of digitally signed messages or vcards that can reveal the identity of the content provider. Applications that automatically attach digital signatures or vcards to messages should be re-configured to disable this functionality when anonymity is desired.

As a means to ensure that information release policy is being properly implemented, organizations can institute operational procedures such as a multi-party review that dictate workflow for products intended for external release. Workflow for certain types of information can incorporate the use of additional reviewers primarily to signal potential anonymity compromises but also to measure compliance with company policy.

Once policy and procedures for anonymizing message content are in place, users must be trained to ensure that they understand and can implement them. Training should be specific to the current policy and procedures and repeated on a regular basis to reduce

the chances of a compromise. One aspect of training that should be emphasized relates to the use of subtle clues within information that could reveal too much about its source or subject. Users should be advised to avoid the use of particular wording, spelling or grammar, or particular styles of writing that could uniquely identify the information provider to others in the group. Issues of writing style are particularly problematic in smaller groups where members may know each other well and may be able to easily recognize familiar patterns of language. Articles [7] and [10] discuss these issues in more detail.

6.2 Information Pre-Processing

One approach to providing anonymity in message content is to rely on information providers to "do the right thing" when preparing products for release. This approach assumes that users adhere to policy and procedures and that they have been properly trained. Other approaches to preserve anonymity that can be used alone or in combination with "doing the right thing", involve bringing structure to the messages either before or after their creation. Structured forms can eliminate the use of "free text" and thus rigidly enforce compliance to message content policies. Structure can also be applied to messages after they have been created by processing them prior to their cryptographic packaging and communication over public networks. This pre-processing is most often handled via the use of software that provides automated information review and sanitization. One example of this is the use of stylesheets that can be applied to XML data to strip or modify information before it is released. Filters can be applied to other types of data as well to ensure that sensitive content is removed. Message filtering could be tailored to satisfy site-specific security policy and modified to accommodate changing views on how much information is to be shared within a group. In some cases, more detailed information could be revealed to enable greater precision in information analysis.

More sophisticated techniques such as Bayesian filtering could also be used to reduce the likelihood of compromised anonymity due to writing style or subtle clues that may be mined from information intended for release. These techniques can provide greater guarantees of compliance with information release policy as well as the spirit of that policy. Security administrators should be provided with easy-to-use tools for tailoring information filters, enabling the organization to adapt to changes in site security policy and evolving information sharing needs. Modifications to information filters should be accompanied by changes to information release policy and procedures as well as updated user training.

7 Applications

The ability to share information electronically in an anonymous yet authenticated way has several important applications. Requirements for sharing information exist across numerous infrastructure sectors, such as gas and oil, electric, and banking and finance. In these instances, sharing information safely occurs via trusted parties, through a membership forum or verbal meetings. Both U.S. and European infrastructure forums share information with a vetted group through existing guidelines on what information

can be divulged. If information is shared electronically, typically the hosting organization bears the burden of protecting the information from misuse or disclosure. These underlying factors often have an effect on what information an organization shares. A burden of trust can affect the entire sharing process and create obstacles for the sharing parties as well as the hosting forum.

Given that the standard mechanism for sharing information lacks a robust form of anonymity, an electronic solution that ensures authentication and affirms anonymity is of particular value. Using the technologies and processes described in this paper removes the burden of information protection by the hosting organization, the need for guidelines about disclosure, and it fosters sharing in an electronic space with identity protection. This approach facilitates sharing by providing added assurance to members of the sharing organizations. Given the assurance of anonymity, members may be willing to share more useful information that can be aggregated to enhance critical infrastructure security as a whole. Likewise an electronic mechanism creates the capability to share information instantaneously, rather than waiting to share critical information at in-person forums. For example, timeliness may be crucial when sharing information that pertains to critical infrastructure protection. An anonymous authentication solution has many benefits that foster the overall sharing and dissemination of information.

Anonymous, authenticated communication has been applied to the I3P's Cross-Domain Information Sharing (CDIS) solution, prototyped in 2006. This application provides a mechanism to share security incident information with the assurance of authentication and no electronic traceability to the identity of the sharing organization. This is of particular importance to the prototype audience, oil and gas industry owners and operators. This highly competitive industry requires a high level of assurance to motivate sharing. In this instance, a sector Center would be established to handle incoming incident reports from industry participants.

The industry would utilize an input form to report characteristics of a security incident and submission to the Center would be facilitated using anonymous, authenticated communication. The incident data would undergo correlation as well as statistical and trend analysis. Aggregated information and analysis would then be distributed to industry members through alerts, status, and lessons learned to allow them to prepare, recognize, and react to common security events. Asset owners, vendors, and the center could optionally interact with other information sharing centers at their discretion. The CDIS prototype illustrates a concept for information sharing and providing assistance to industry in identifying coordinated attacks and protecting the critical infrastructure while maintaining user anonymity. The authentication requirements in submission of data increase the trustworthiness of the information. Shared data from trusted members provides a higher confidence of accuracy in detecting security events.

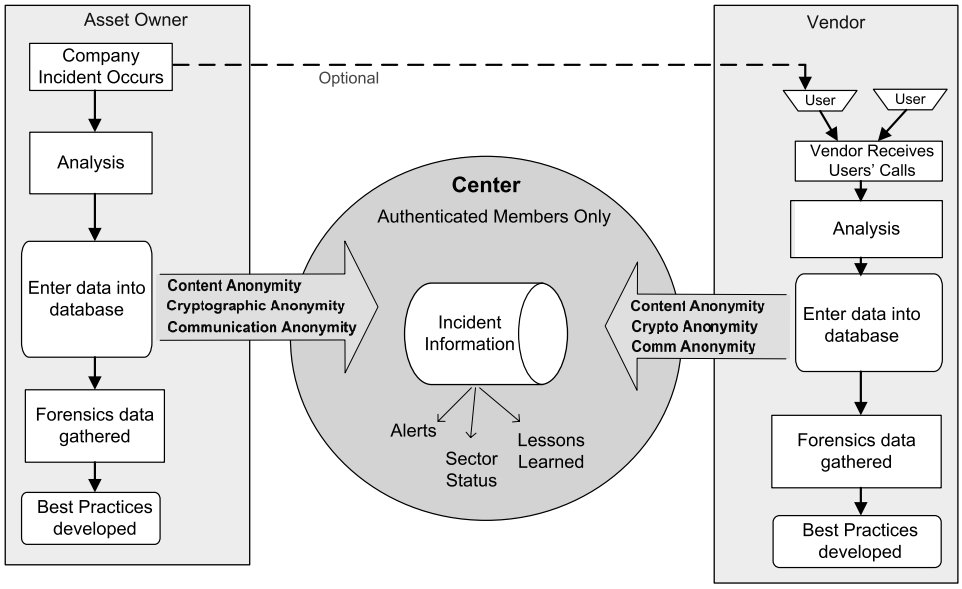

Figure 3 – Application of anonymous, authenticated communication to SCADA/PCS information sharing.

Figure 3 presents a notional view of information sharing within the SCADA / PCS community where asset owners and vendors can participate in sharing of incident information. In such a system, an asset owner might interact with a vendor to determine the cause of an incident or the appropriate response. Since vendors communicate with multiple asset owners, they might be the first to detect a coordinated attack. Anonymous authentication is a central feature, or facilitator, in the CDIS concept that separates the prototype from other information sharing mechanisms.

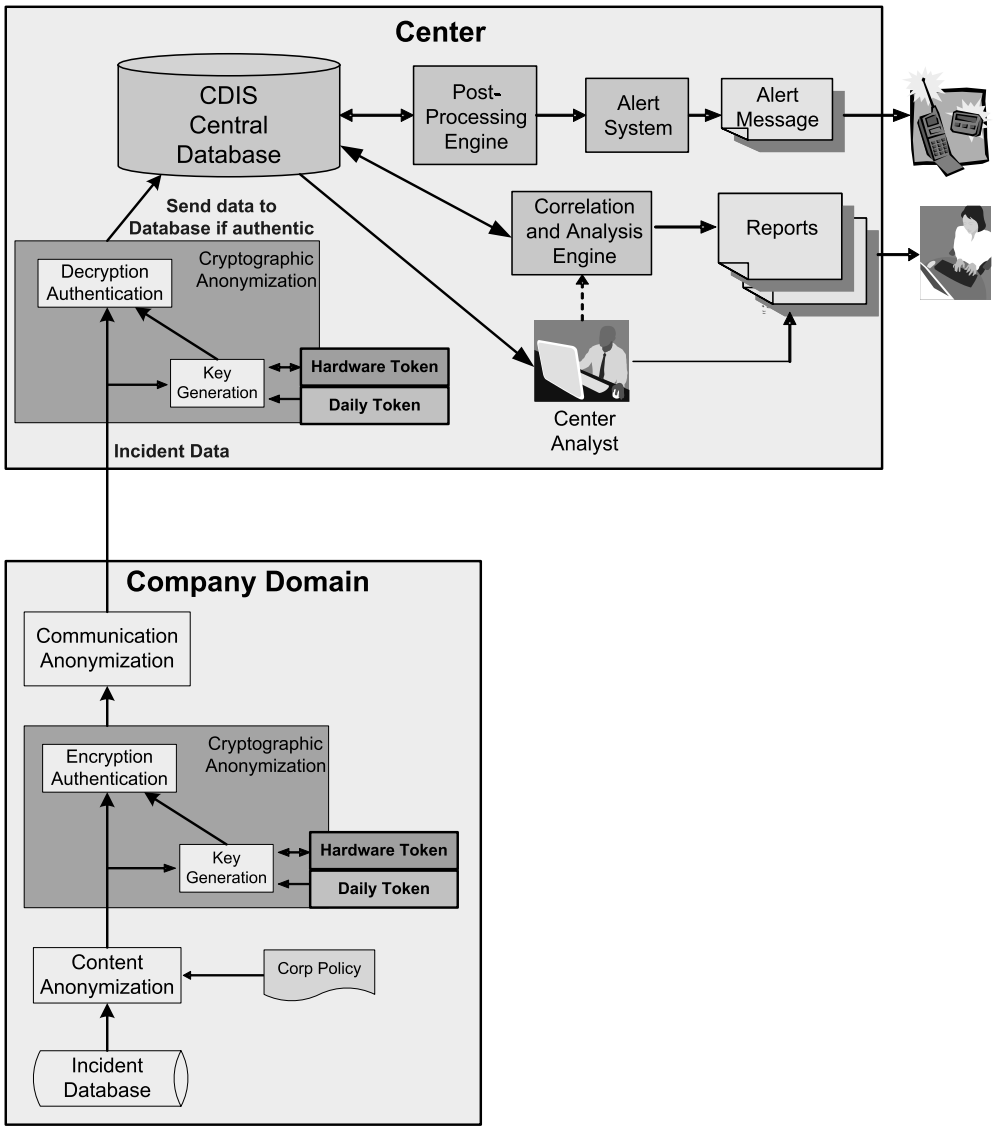

Figure 4 – Anonymous Authentication in a Prototype Cross Domain Information Sharing Architecture.

Figure 4 illustrates the usage of the elements of anonymous, authenticated communications in the context of the I3P CDIS prototype architecture. Key aspects of this usage include the following.

- *All domains use the same key material.* – During initial setup of the anonymous, authenticated communication system, hardware token with random numbers are acquired from the Center.

- *The key material changes daily to avoid proliferation* – The Center randomly generates and distributes a daily token. The daily token is encrypted with each domain's public key, sent to each domain, and the hash of all encryptions is publicly posted by the Center. Therefore, each member can verify the integrity of the daily token.
- *New encryption and authentication keys are generated for each message* – The message sender encrypts and authenticates the message and sends it to the Center along with an index to the hardware token.
- *New decryption and authentication keys are generated for each received* – Using an index from the message sender, the decryption key is generated from the common key material. The recipient can decrypt the message verify that it came from an authentic user, but doesn't know which one.
- *The Center can revoke a message without revealing the sender's identity.* (not implemented in the CDIS prototype).

This example illustrates how the use of cryptography-based anonymous authentication solution can facilitate information sharing through protection of identity and data. This concept can be applied to many situations in which data must be shared with other parties, but a need for trusted information from trusted partners exists. At the core of these situations is the transfer of information. Other applications might include electronic voting, site access control, and other critical infrastructure such as transportation.

8 Conclusions

The initial motivation for this research was Presidential Directive 63 which defined a scheme where private industry would send important information to a central organization, which would forward the information to a federal organization. The participants needed a way to provide information without revealing their own identity or compromising their proprietary property. The schemes described in this paper have been developed with these considerations in mind.

SCADA and PCS are attractive targets for disabling the national critical infrastructure. Communication between stakeholders of an infrastructure community and the aggregation and analysis of the shared information can offer security benefits to the community as a whole and each member as well. However, sharing of meaningful information is often not worth the risk of identifying the source. Complete, trustworthy anonymization of shared information is a complex problem that must address the original contents of the information, its packaging for delivery to other members or to a central repository, and the delivery itself across a public communication network. We provide a technical discussion and describe the critical elements of a complete solution for anonymous but authenticated information sharing with data integrity and confidentiality protection. A complete solution also provides a means of revoking messages without removing their anonymity. This anonymous, authenticated communication system is part of a cross-domain information sharing

demonstration system developed for the oil and gas industry. However, the technical approach described in this paper is applicable to many other industries in need of secure information sharing.

About the Authors – Tim Draelos, Annie McIntyre, and Rich Schroeppel are all members of the technical staff and William Neumann is a Ph.D. student intern at Sandia National Laboratories. Tim Draelos conducts research and development in cryptographic applications, implementations, and algorithms as well as in the use of machine learning algorithms for security applications. Tim has led and participated in several R&D projects in SCADA security. He holds a Ph.D. in electrical engineering from the University of New Mexico and is a member of the IEEE. Annie McIntyre's primary areas of research include threats, vulnerabilities, and protection of critical infrastructure under the I3P Program and leading security metrics efforts under the National SCADA Test Bed. She holds a Bachelor of Science from New Mexico Tech, a Master of Science from Troy State University, and is a member of the American Association of Petroleum Geologists. William Neumann's research consists of cryptographic implementations, analysis, and obfuscation. Rich Schroeppel is a mathematician involved in research and development of cryptographic algorithms, analysis, and implementations. He has worked in the field for over 30 years, developing methods for factoring integers, establishing the practicality of software elliptic curve algorithms, and contributing the Hasty Pudding Cipher to the NIST block cipher competition.

Christine Eliopoulos is a Principal INFOSEC Engineer at the MITRE Corporation. Her primary areas of research include cross-domain collaboration and information sharing. She has supported a number of efforts to architect and deploy information sharing technologies securely. Ms. Eliopoulos obtained a Bachelor of Science degree in Computer Science from Merrimack College and a Master of Science degree in Computer Science from Boston University.

The authors wish to thank Keith Ibarguen of the MITRE Corporation for his excellent reviews of the paper and helpful suggestions during its writing.

References

[1] C. Beaver, R. Schroeppel, and L. Snyder, *A Design for Anonymous, Authenticated Information Sharing*, from the Proc. 2001 IEEE Workshop on Information Assurance and Security.

[2] D. Boneh and M. Franklin, *Anonymous Authentication with Subset Queries,* Proceedings of the 6th ACM conference on Computer and Communications Security, pp. 113-119, 1999.

[3] British Columbia Institute of Technology, *Featured Projects: Industrial Security Incident Knowledgebase*, http://www.bcit.ca/appliedresearch/security/projects/knowledge.shtml

[4] D. Chaum, *Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms,* Communications of the ACM, vol. 24, no. 2, pp. 84-88, 1981.

[5] A. De Santis, G. Di Crescenzo and B. Persiano, *Communication Efficient Anonymous Group Identification,* Proceedings of the ACM Conference on Computer and Communications Security, pp. 73-82, 1998.

[6] R. Dingledine, N. Mathewson, and P. Syverson, *Tor: The Second-Generation Onion Router*, Proceedings of the 13th USENIX Security Symposium, August 2004.

[7] B. Gavish and J. H. Gerdes Jr., *Anonymous Mechanisms in Group Decision Support Systems Communications*, Decision Support Systems, v. 23, no. 4, pp. 297-328, 1998.

[8] I. Goldberg, D. Wagner and E. Brewer, *Privacy-Enhancing Technologies for the Internet,* Proceedings of COMPCON '97, pp. 103-109, 1997.

[9] D. Goldschlag, M. Reed, and P. Syverson, *Hiding Routing Information,* Proceedings of Information Hiding, pp. 137-150, 1996.

[10] S. Hayne, and R. Rice, *Attribution Accuracy When Using Anonymity in Group Support System,* International Journal of Human Computer Studies, v.47, no. 3, pp. 429-452, 1997.

[11] A. Mcintyre, J. Stamp and A. Lanzone, *I3P Preliminary Risk Characterization Report*, I3P Research Report. 2006.

[12] K. Oish, M. Mambo, and E. Okamoto, *Anonymous Public-Key Certificates and their Applications,* IEICE Transactions, vo1.E81A, no. 1, pp. 65-71, 1998.

[13] T. Pedersen, *Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing,* Advances in Cryptology - CRYPTO '91, pp. 129-140.

[14] Process Control Systems Forum, *Control System Security Event Monitoring,* https://www.pcsforum.org/groups/66/

[15] Process Control Systems Forum, *SCADA Threat Statistics,* https://www.pcsforum.org/groups/66/reporting/em2.php

[16] M. K. Reiter and A. Rubin, *Crowds: Anonymity for Web Transactions*, ACM Transactions on Information and Systems Security, pp. 62-92, 1998.

[17] S. Schechter, T. Parnell, and A. Hartemink, *Anonymous Authentication of Membership* in *Dynamic Groups,* Proceedings of Financial Cryptography '99, vol. 1648 LNCS, pp. 184-195.

[18] P. Syverson, D. Goldschlag, and M. Reed, *Anonymous connections and onion routing*, IEEE Journal on Selected Areas in Communication, v. 16, no. 4, pp. 482-494, 1998.

[19] J. Watters and C. Eliopoulos, *Requirements for Cross-domain Information Sharing Within SCADA Environments*, I3P Research Report, 2005.

[20] Cypherpunks, *Anonymous Remailers*, http://cypherpunks.org/remailers/

[21] US Department of Energy Office of Electricity Delivery and Energy Reliability, *Infrastructure Security and Energy Restoration,* http://www.oe.netl.doe.gov/index.aspx

[22] US Department of Homeland Security, *Activities and Programs,* http://www.dhs.gov/xinfoshare/programs/

[23] US Department of Homeland Security, *Fact Sheet: Homeland Security Operations Center (HSOC),* http://www.dhs.gov/xnews/releases/press_release_0456.shtm

[24] US Department of Homeland Security, *Homeland Security Information Network,* http://www.dhs.gov/xinfoshare/programs/gc_1156888108137.shtm

[25] US Department of Homeland Security, *US-CERT About Us,* http://www.us-cert.gov/aboutus.html

[26] US Department of Homeland Security, US Computer Emergency Readiness Team, http://www.us-cert.gov

[27] World Wide/Information Sharing and Analysis Center, About the World Wide ISAC, http://www.wwisac.com/about/