SAND2007-0365C

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

SCADA Security Scientific Symposium

January 25, 2007

Sandia National Laboratories

Research Team

Sandia National Laboratories

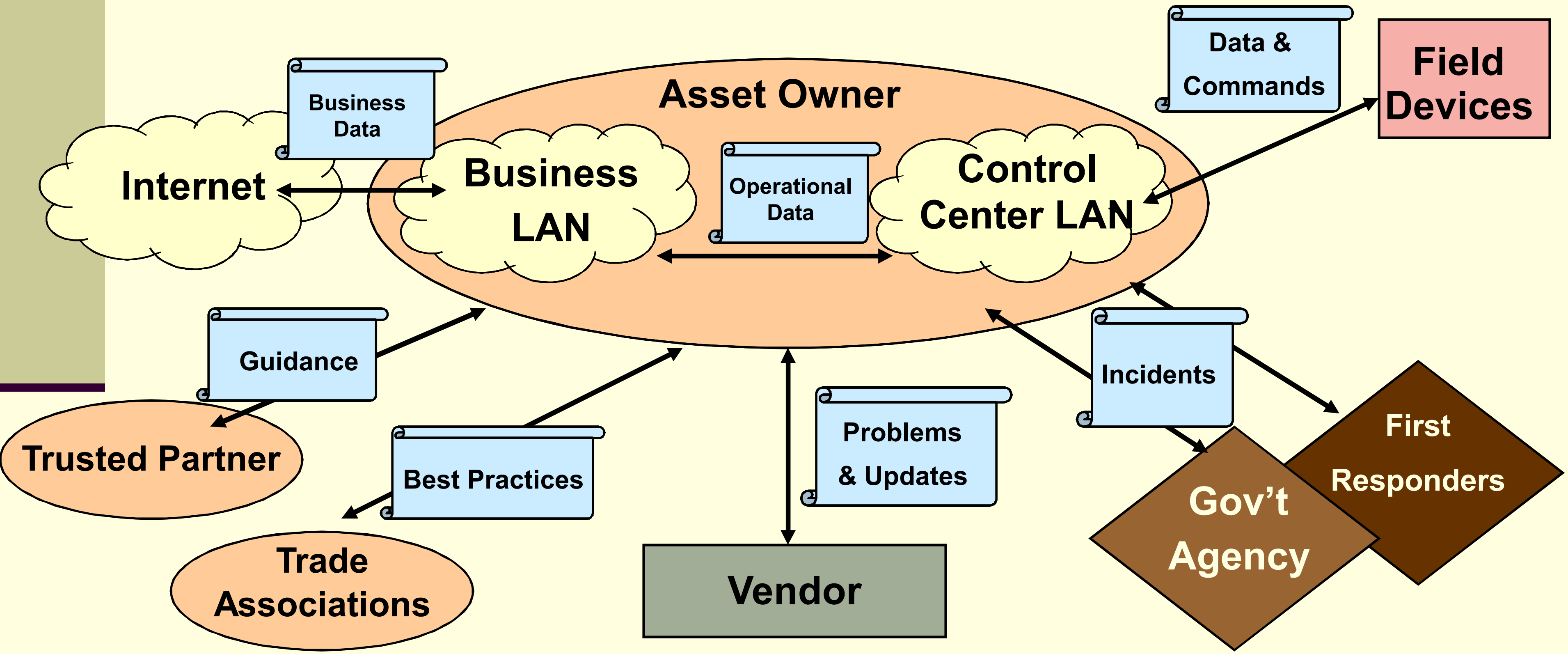
- Tim Draelos
- Annie McIntyre
- William Neumann
- Rich Schroeppel

MITRE

- Chris Eliopoulos

Problem Space

- Cross-Domain Information Sharing (CDIS)
  - e.g., SCADA / PCS Community



Information Sharing – Why Share?

- Protect against, better understand, & respond to
  - Cyber, personnel, physical security threats
- Info sharing among industry asset owners and vendors could help prevent, detect, or counter these threats.
  - Early detection of coordinated attacks
- Common community interest → Healthy industry
  - Tool to protect assets and ensure uninterrupted operations and service
  - ***Cost of inaction*** can be > ***Cost of prevention***
    - downtime, public confidence, equipment repair

Information Sharing – What to Share?

- System status
  - Equipment failures
- Surveillance information
- Incident information
  - Recent attacks, effects, and actions
- Security solutions
  - Secure configurations
  - Best practices

Existing Information Sharing Efforts

- Homeland Security Information Network
  - HSIN
- US Computer Emergency Readiness Team
  - US-CERT
- Infrastructure, Security, and Energy Restoration
  - ISER
- Energy Information Sharing Analysis Center
  - ISAC
- Industrial Security Incident Database
  - ISID
- Control System Security Event Monitoring Working Group

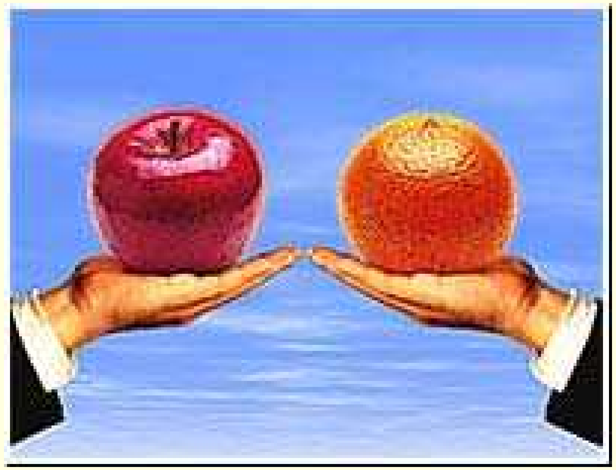
Issues with Existing Information Sharing Efforts

- What is the motivation for use of these systems?
- Is a secure infrastructure in place?
- What are the Control of shared information?
- Where does the information end up?
- Who’s in charge?

Challenges to Information Sharing

- Establishing trust

- ROI
  - Better protection, preparedness

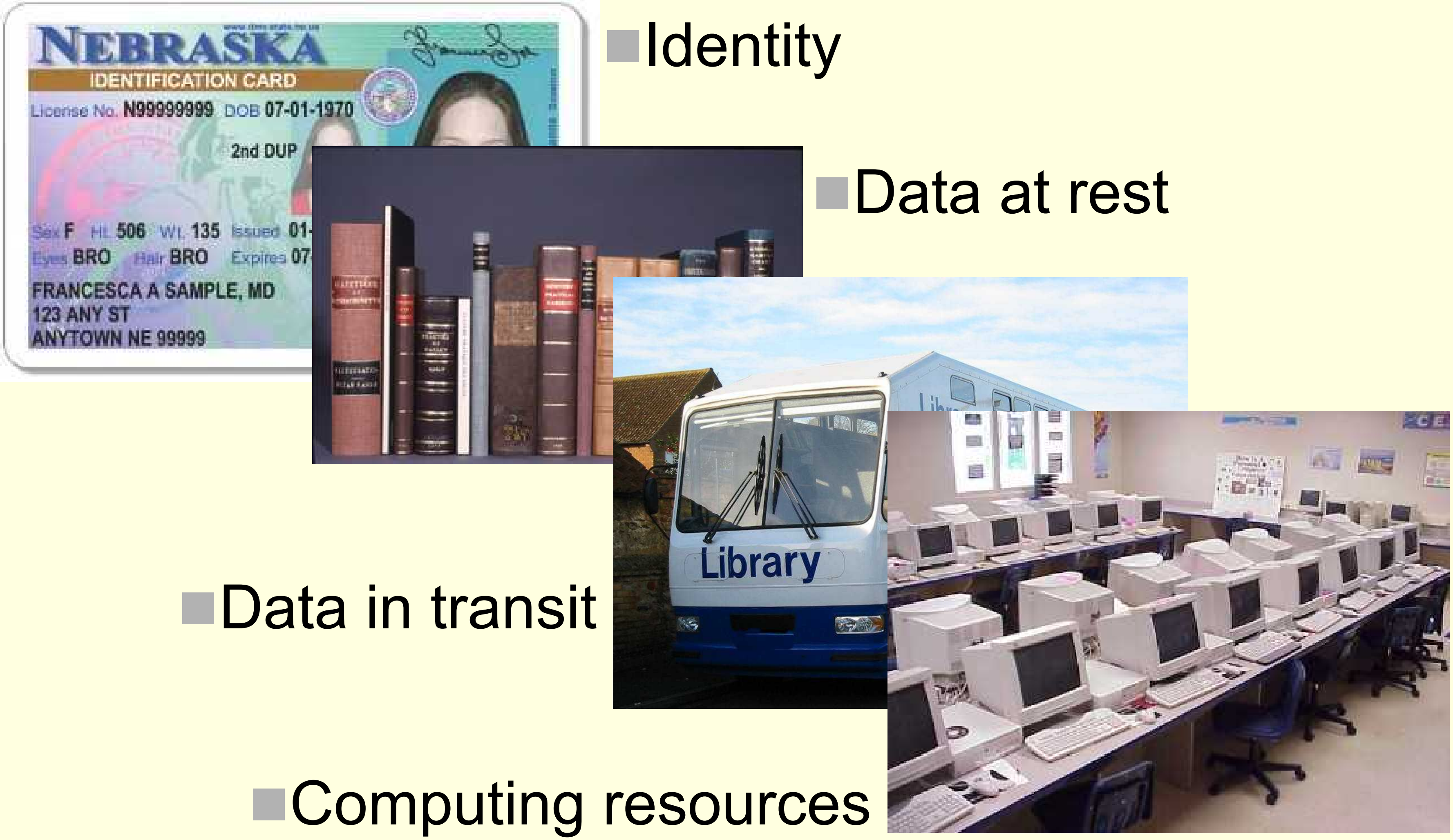


- Standardized data

Challenges to Information Sharing:

Information Protection

- Identity
- Data at rest
- Data in transit
- Computing resources


NEBRASKA
IDENTIFICATION CARD
License No. N99999999 DOB 07-01-1970
2nd DUP
Sex F Ht. 506 Wt. 135 Issued 01-
Eyes BRO Hair BRO Expires 07-
FRANCESCA A SAMPLE, MD
123 ANY ST
ANYTOWN NE 99999
Library

Attacks on Information Sharing Systems

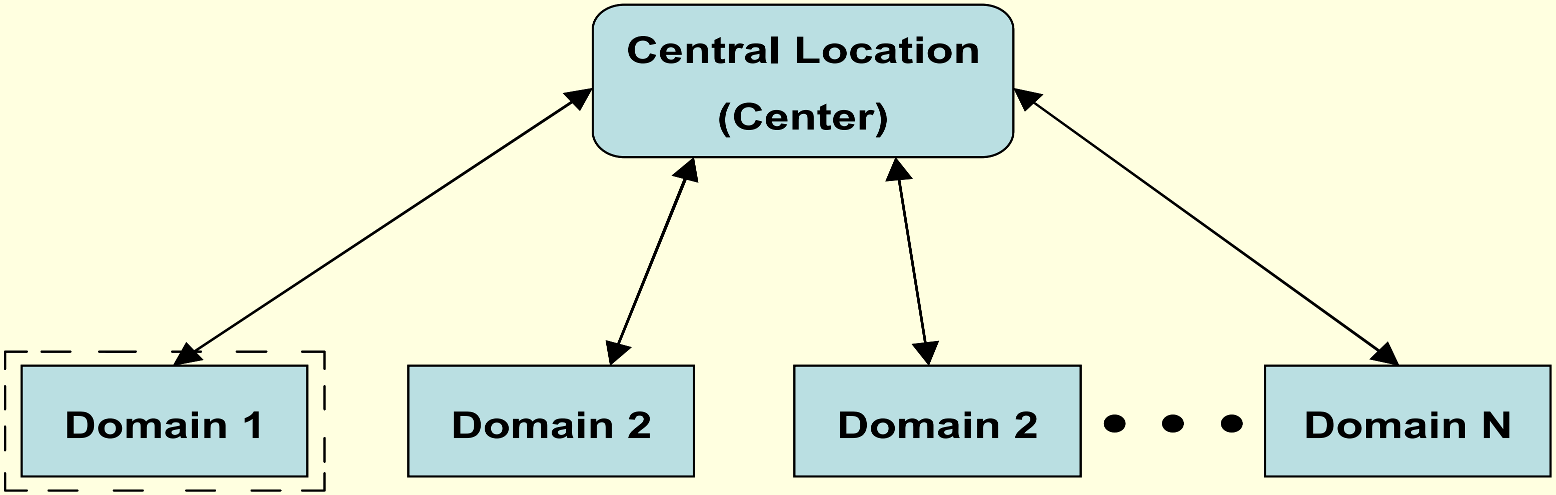
- Systems without anonymity protection
  - Trace communication back to its source
  - Read contents to determine the author
- Systems with anonymity protection
  - Forge an incident report, Spam, Denial-of-Service
    - Motivation for authentication
  - Submit bogus incidents to bias/corrupt database, analysis
    - Motivation for anonymity revocation with reply tokens

Design Goals of Anonymous, Authenticated Communication

- Anonymity of the information provider
- Anonymous network communication paths
- Authentication of the source
- Anonymity, confidentiality, & integrity of the data
- Protection against system abuse by insiders

- Our Proposal
  - **Cryptographic** Anonymization
  - **Communication** Anonymization
  - **Content** Anonymization

Communication Model



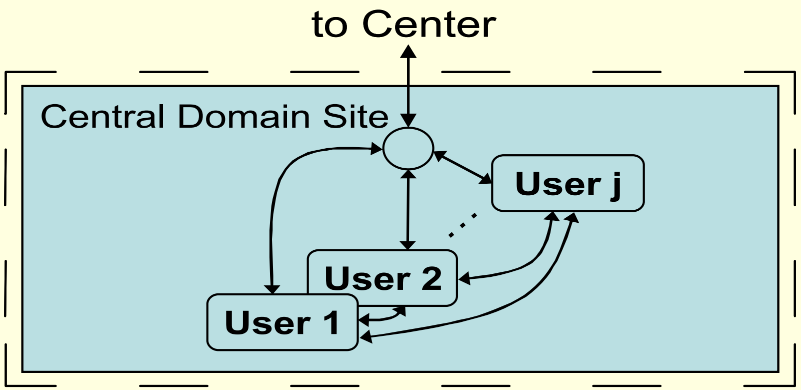
Central Location (Center)

Domain 1

Domain 2

Domain 2

Domain N



to Center

Central Domain Site

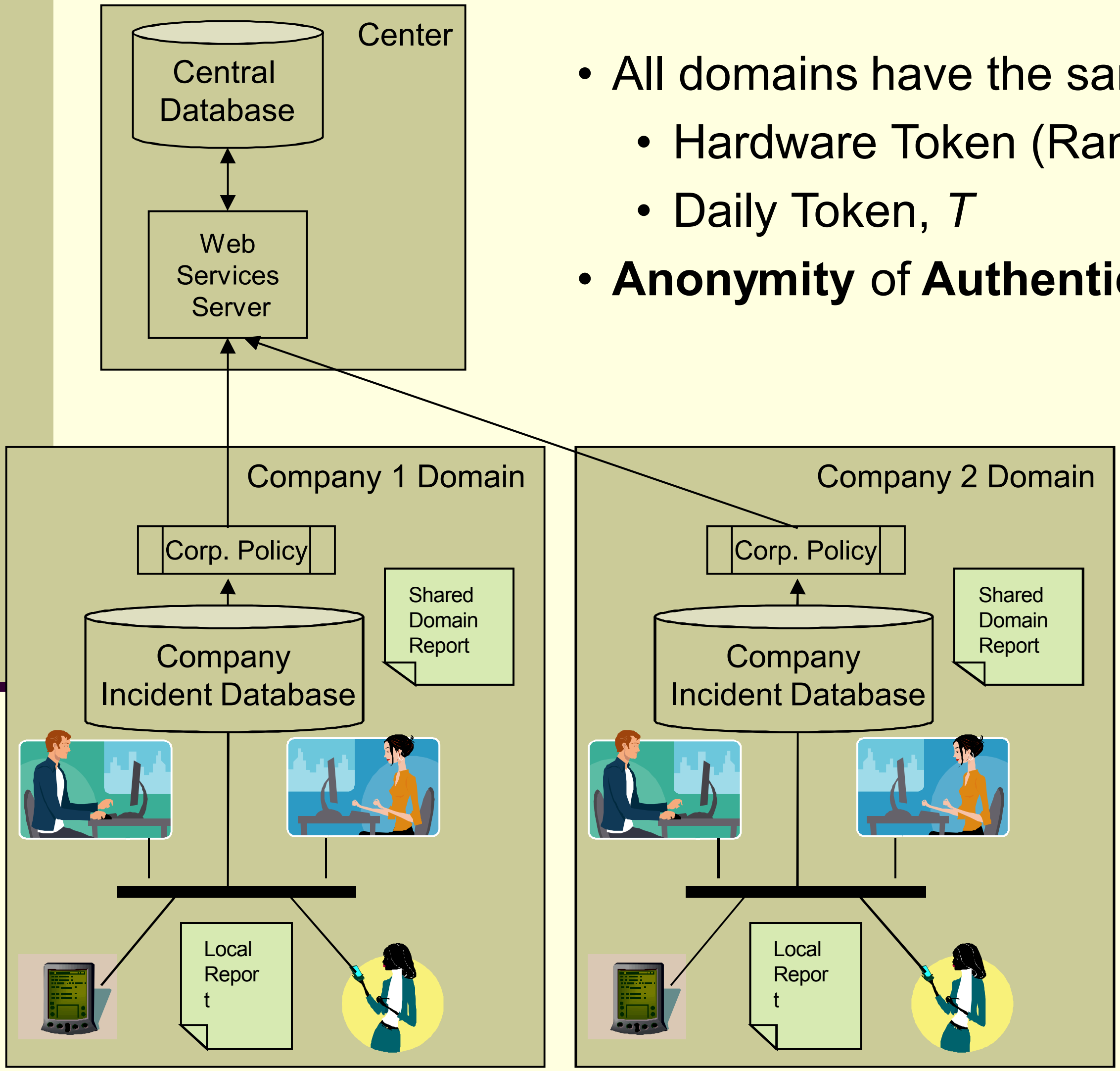
User j

User 2

User 1

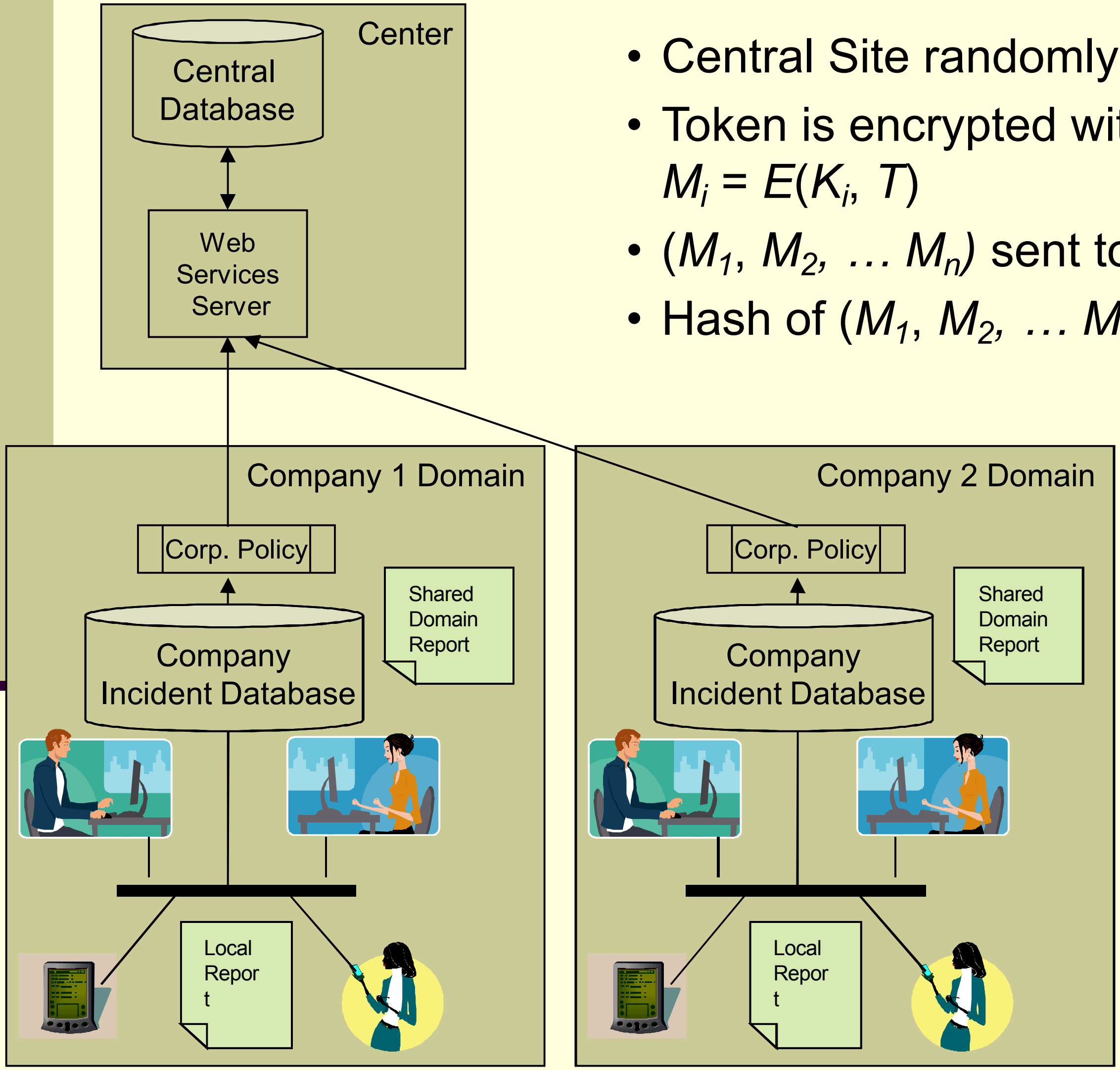
Cryptographic Anonymization: Key Concepts

- All domains have the same key material
  - Hardware Token (Random Number CD)
  - Daily Token, *T*
- **Anonymity** of **Authenticated** users



- Each communicated message uses a different key
- Key material changes daily to avoid proliferation
  - Old keys are useless

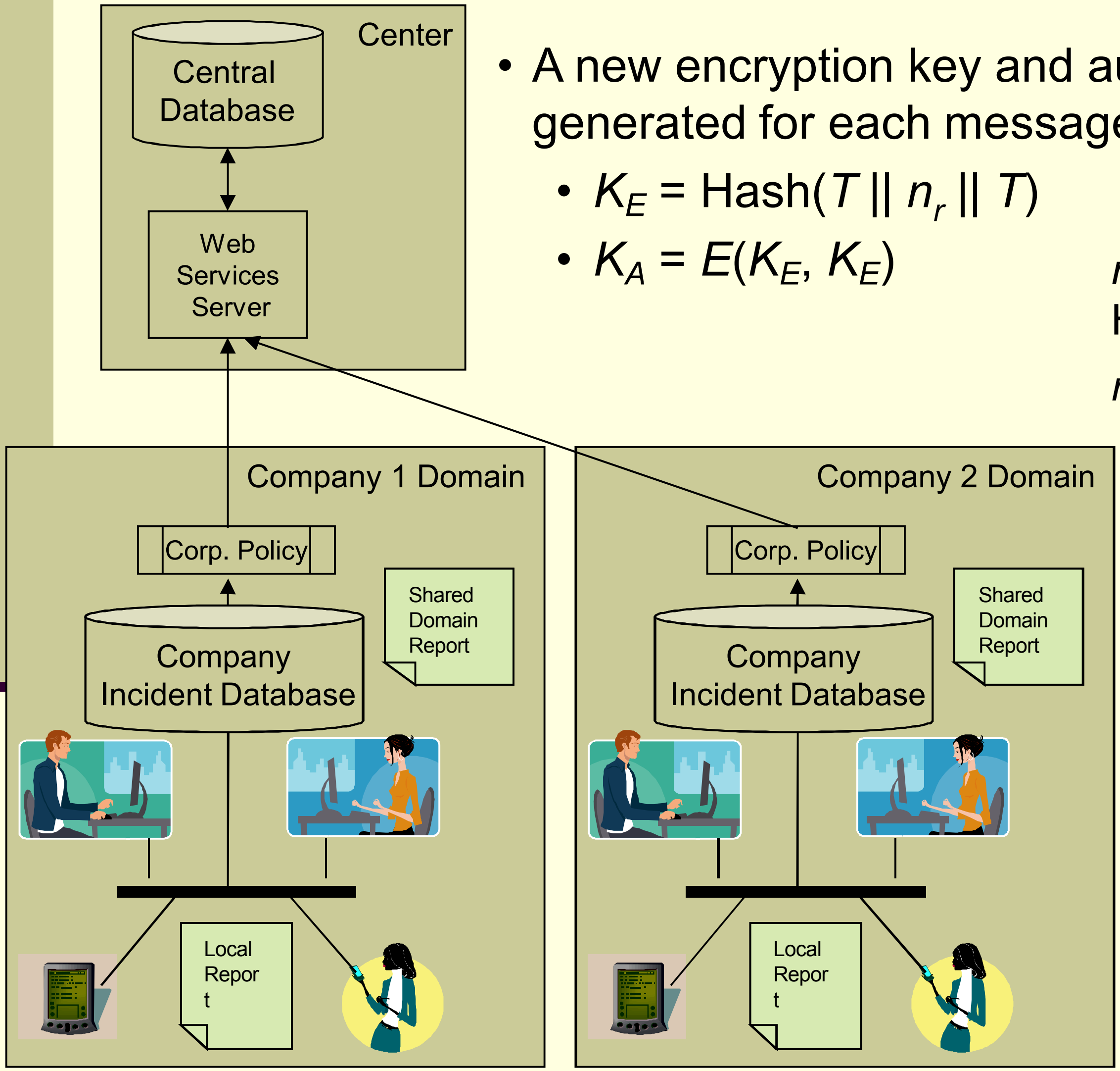
Daily Token Generation and Verification



- Central Site randomly generates a daily token, *T*
- Token is encrypted with each domain's public key, $M_i = E(K_i, T)$
- $(M_1, M_2, \ldots M_n)$ sent to each domain
- Hash of $(M_1, M_2, \ldots M_n)$ published in public location

- Each domain *i*
  - Verifies public hash of $(M_1, M_2, \ldots M_n)$
  - Decrypts daily token
    - $T = D(K_i, M_i)$
  - Verifies each M_j for all i ≠ j
    - $M_j = E(K_j, T)$?

Key Generation and Encryption

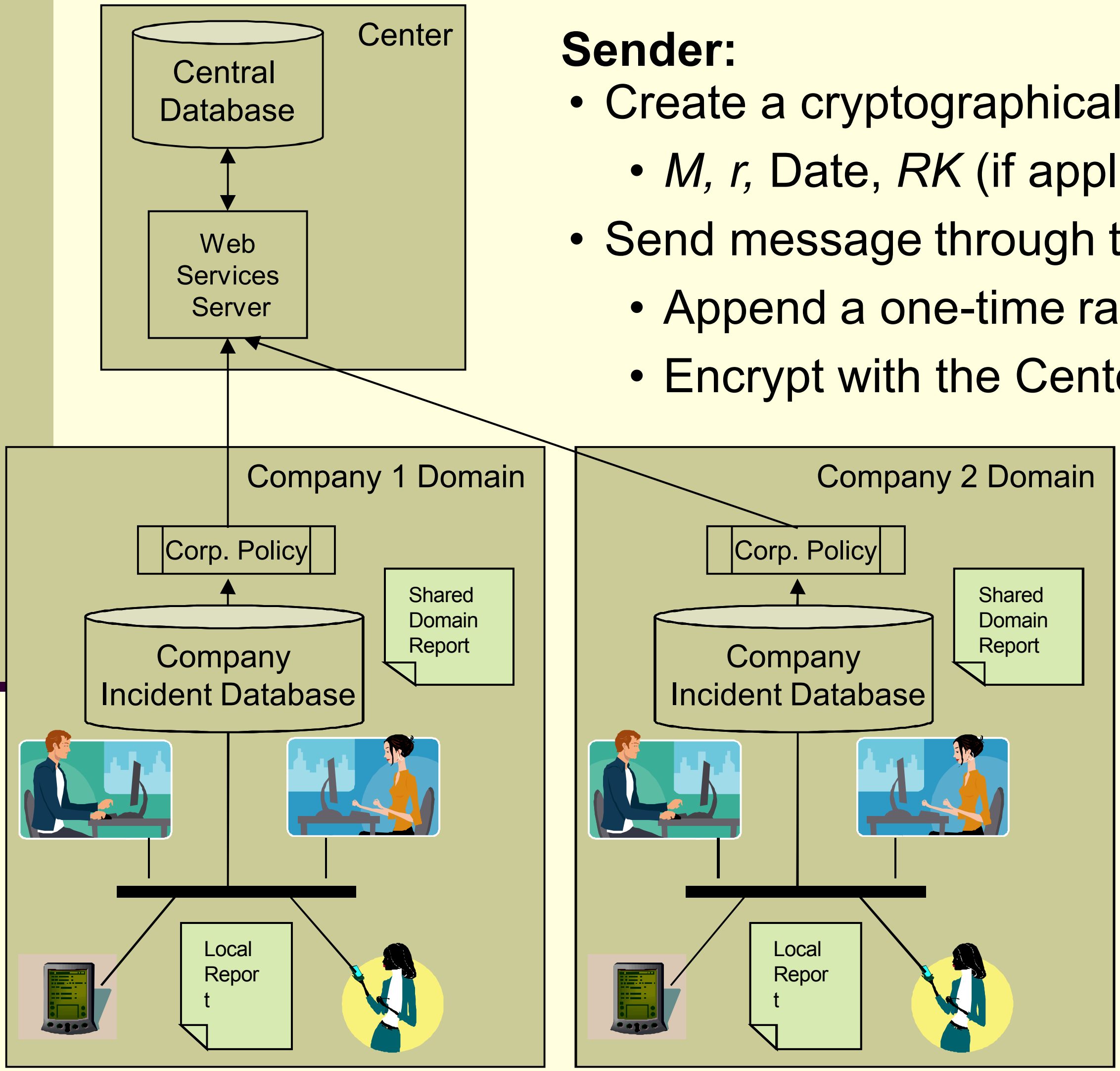


- A new encryption key and authentication key are generated for each message using common key material
  - K_E = Hash(T || n_r || T)
  - $K_A = E(K_E, K_E)$

n_r = Random # from HW Token
Hash(n_r) must match part of Hash(T)
r = Index into the HW Token

- Incident / Alarm message is encrypted and authenticated
  - $M = E(K_E, message)$
  - $A = MAC(K_A, M)$ or $MAC(K_A, A \,||\, E(K_E, RK))$
    - if using anonymous message revocation

Message Sending & Receiving



Sender:

- Create a cryptographically protected file, CP-MSG
  - *M, r,* Date, *RK* (if applicable), *A*
- Send message through the anonymous comm channel
  - Append a one-time random reply key, *RK*
  - Encrypt with the Center's public key

M - Encrypted message

r - Index into the HW Token

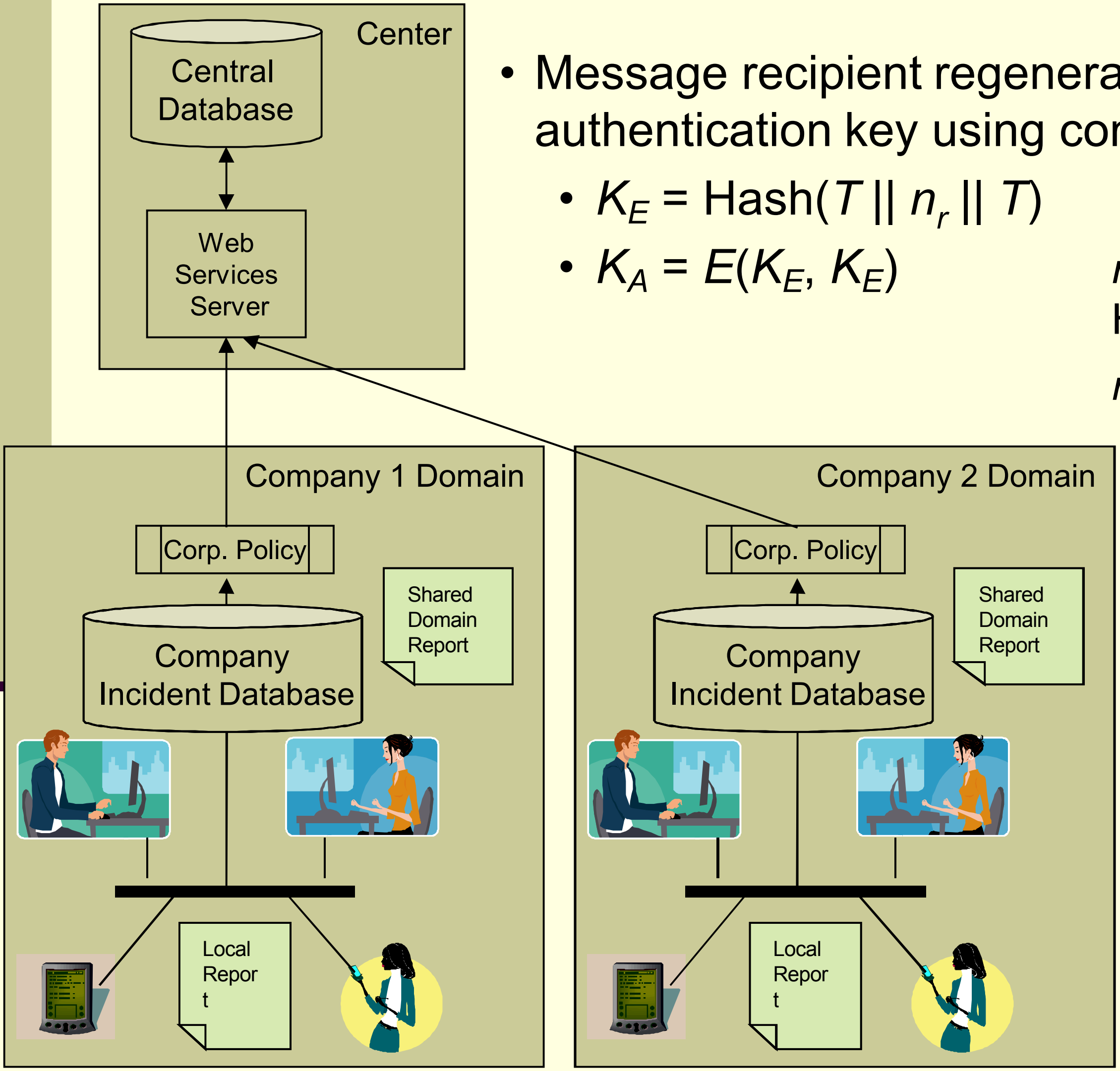
RK - Encrypted reply key

A - MAC of the encrypted message

Receiver:

- Decrypt (CP-MSG, *RK)* using Center's private key

Key Generation and Decryption



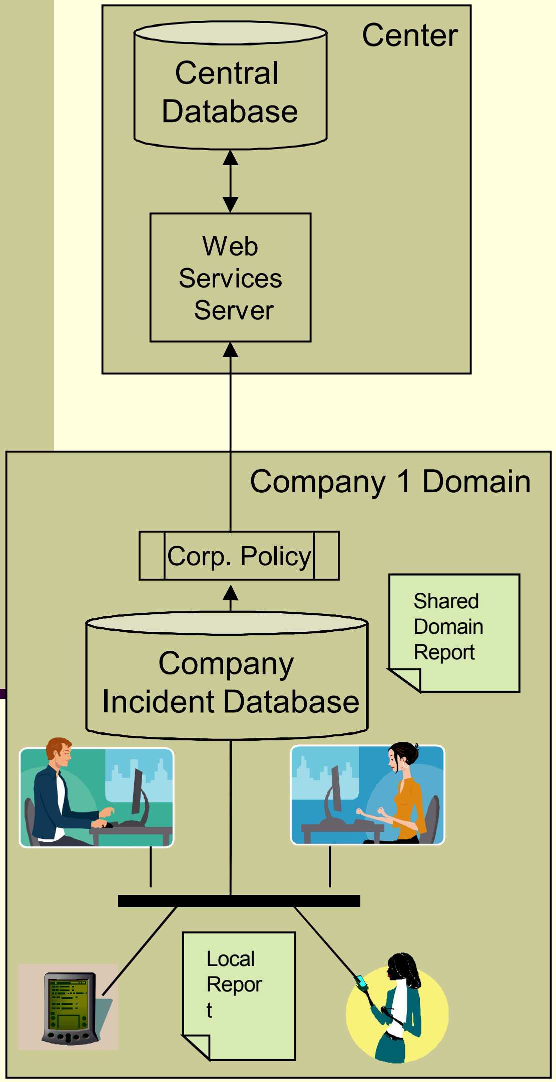
- Message recipient regenerates the encryption key and authentication key using common key material
  - K_E = Hash(T || n_r || T)
  - K_A = $E(K_E, K_E)$

n_r = Random # from HW Token
Hash(n_r) must match part of Hash(T)

r = Index into the HW Token

- Recipient can verify that the message came from an authentic user, but doesn't know which one
- If $MAC(K_A, M) = A$, decrypt the message $D(K_E, M)$.

Anonymous Message Revocation

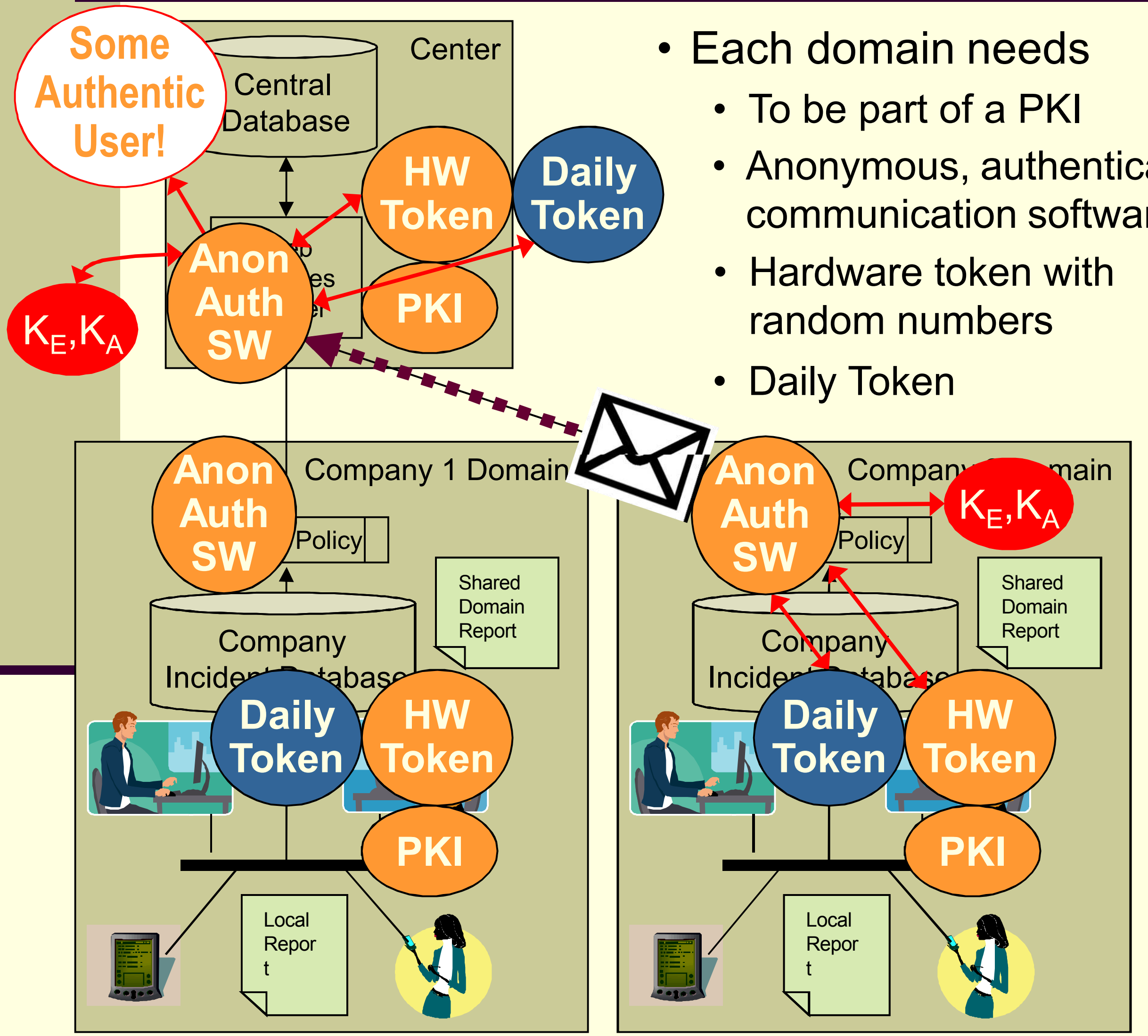


- Center (message recipient)
  - Generates a random 128-bit revocation token, *RT*
  - Encrypts token and message using the reply key, *RK*
    - $V = E(RT \,||\, M, RK)$
  - Sends *V* back to the message originator via anonymous comm channel or via a public site
- Domain receives the reply and sends a copy of the message to domain committee members
- Committee retrieves copy of *RK*, checks the authentication of the message, and decrypts the message if valid.
- The domain committee decides if the message is legitimate. If so, it does nothing, if not, it sends the revocation token, *RT*, back to the Center.

Anonymous, Authenticated Communication

Hash Of Daily Token

Hash of HW Token



- Each domain needs
  - To be part of a PKI
  - Anonymous, authenticated communication software
  - Hardware token with random numbers
  - Daily Token

- Information Sender
  - Accesses HW & Daily tokens
    - to generate symmetric keys
  - Encrypts and packages information with authentication tag
  - Sends package to Center

- Information Receiver
  - Accesses HW & Daily tokens
    - to regenerate symmetric keys
  - Verifies the authentication tag of the received information

    AND
  - Decrypts the information

Communication Anonymization

- Cryptographic anonymous authentication is not enough
- An observer of the communication network might trace the path of information from the sending Domain to the Center
- Need to anonymize network communication between senders and the Center
  - We present multiple options
- All anonymous communication options can use our cryptographic anonymous authentication protocols

Anonymous Communication Channels

- Anonymous remailers
  - Simple
  - Mixmaster
- Randomized proxy networks
  - Crowds
- Onion routers
  - Tor

Anonymous Remailers

- Provide anonymous e-mail channels

- Basic functionality (Type 0)
  - Receive message
  - Strip header information
  - Forward to recipient
- Provides sender anonymity but no more

Anonymous Remailers

- Type 1 remailers offer more services
  - Messages sent through small network
  - Each hop re-encrypts the message
  - Messages passed to next hop only after N messages have been received
- Provides sender anonymity and some sender/receiver-unlinkability
- Vulnerable to replay attacks and advanced traffic analysis

Anonymous Remailers

- Mixmaster (Type 2) remailers offer even more protection
- Made up of multiple nodes
  - Pad and re-encrypt message at each node
  - Random delays added at each node
  - Random path through the mix network
- Greatly improves traffic analysis resilience
- May offer reply e-mail services via pseudonyms and logging

Anonymous Remailer Weaknesses

- Mixmasters provide strong anonymity and unlinkability properties, but are only designed to handle single-message e-mail style traffic
  - Latency can be high (reasonable for e-mail)
  - Reply capabilities require the servers to maintain a log of pseudonyms
    - Can be a point of attack

- Onion routers were designed to solve these problems

Onion Routers

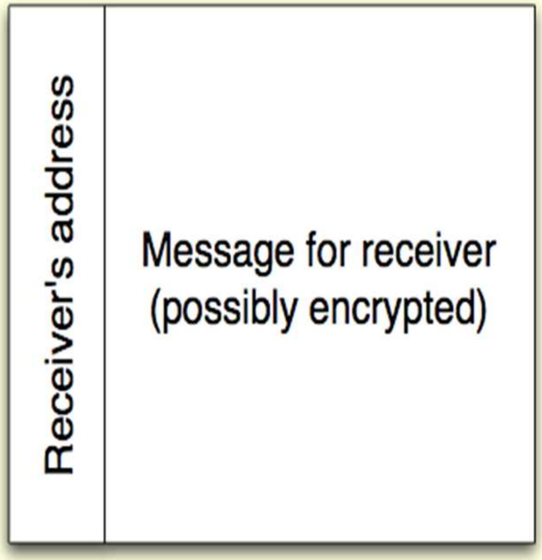
- Much like a Mixmaster network for arbitrary network traffic
  - Employs multiple routers
  - Every packet is stripped of identifying information from the sender
  - Every packet is padded to a uniform size and re-encrypted at each hop in the network
  - Packets are randomly delayed at each hop
  - Random paths taken through the routers

Onion Routers

- At the sender, each packet of the message is wrapped in an “onion” and sent into the router network
- The onion is a layered data structure representing the path the packet will take through the routers to the destination
- The sender randomly chooses this path

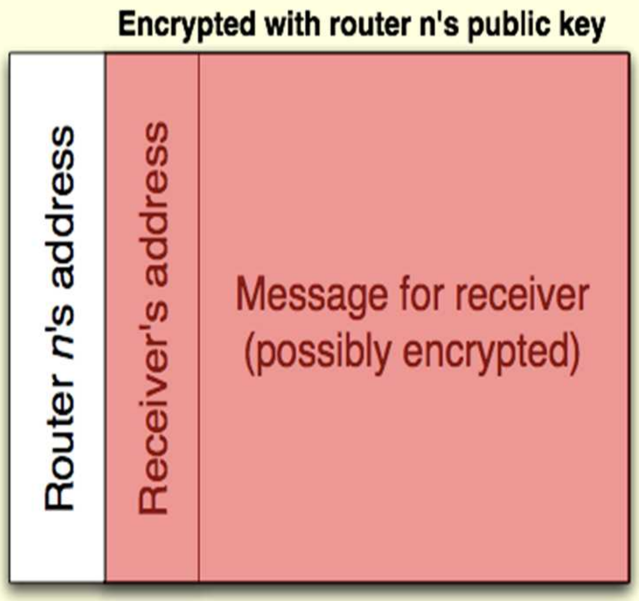
Building an Onion

- The onion creation begins by appending the receiver’s IP address to the possibly encrypted message



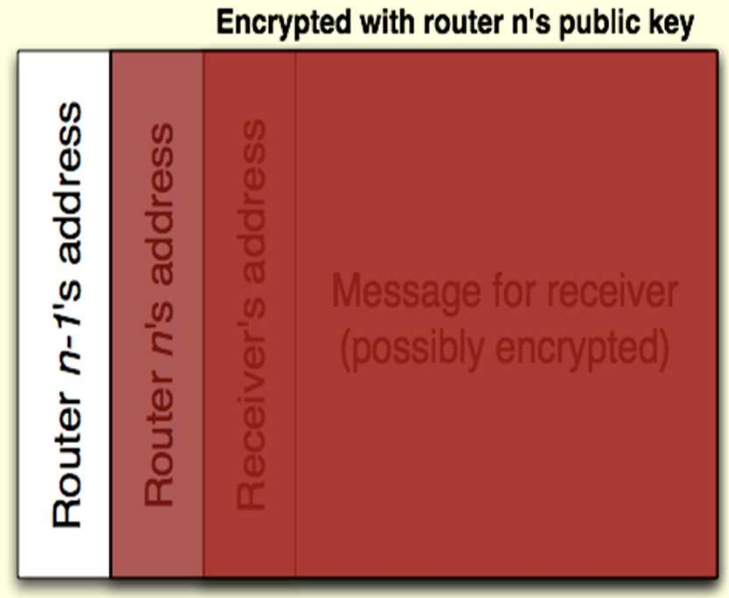
Building an Onion

- This pair is then encrypted with the public key of the final router in the path, and this router's IP address is prepended



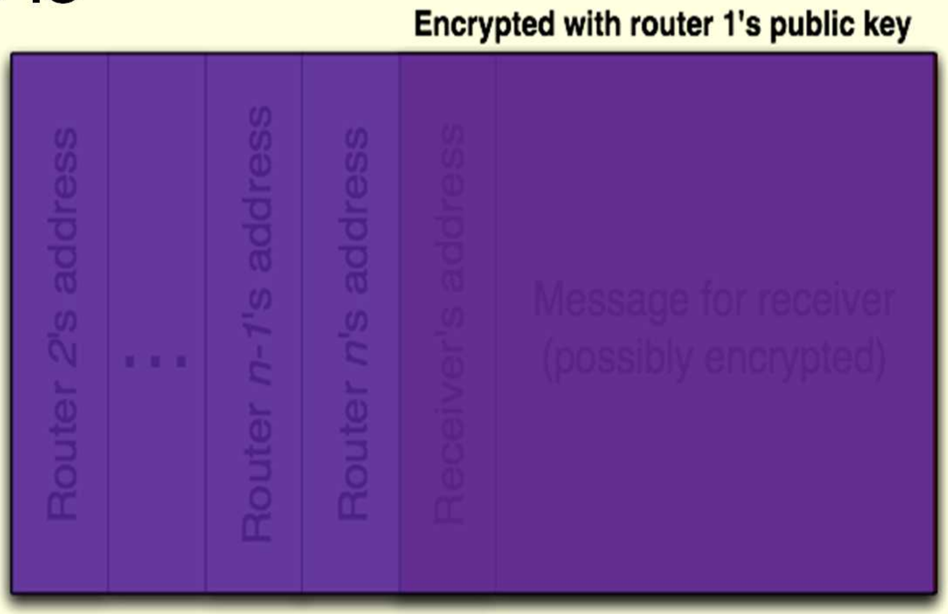
Building an Onion

- This construct is likewise encrypted with the public key of the penultimate router in the path, and its address is prepended



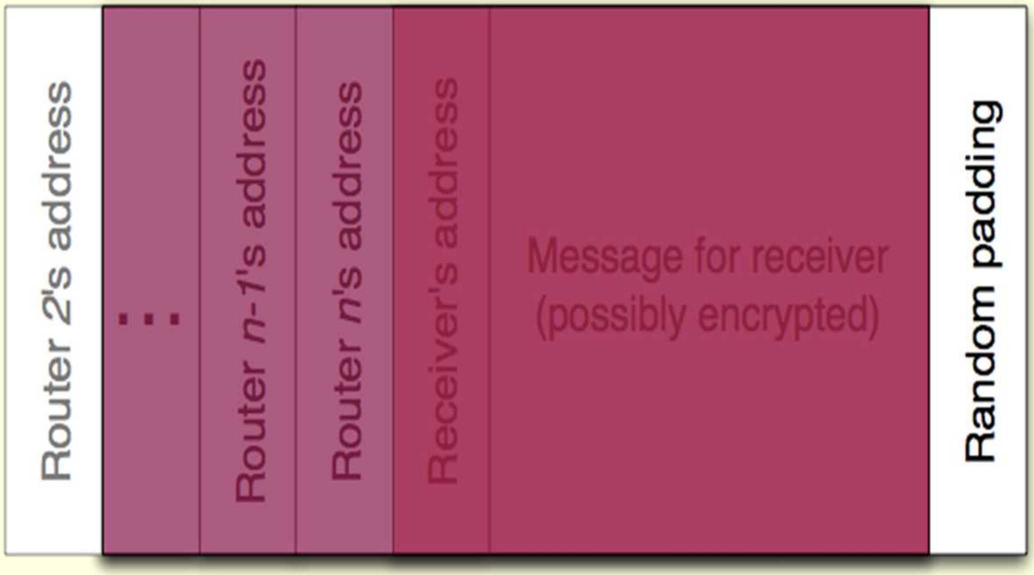
Building an Onion

- The layered encryption continues until the address of the second router in the path (and the associated data) is obscured by the first router's public key
- This onion is sent to the first router



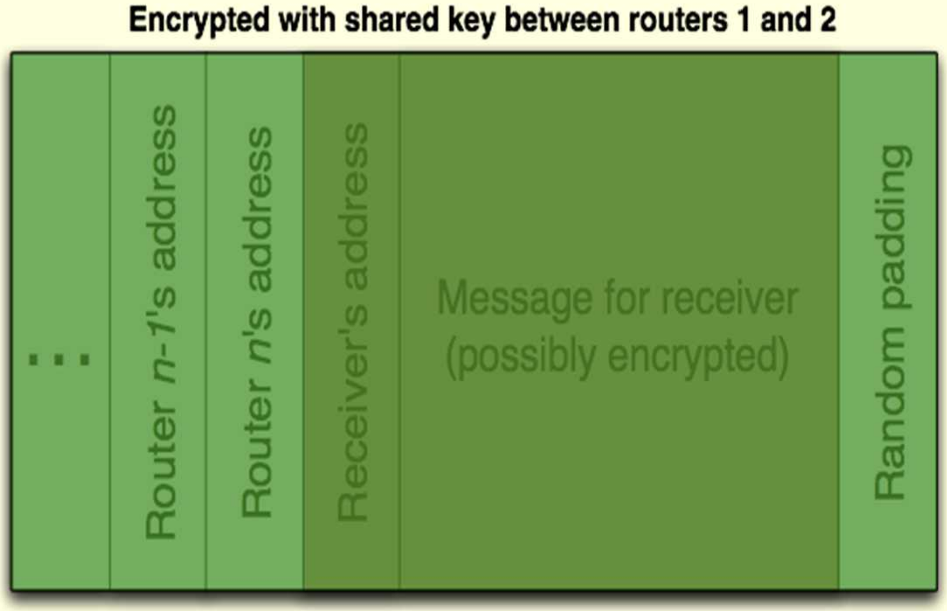
Forwarding an Onion

- The first router decrypts the onion with its private key, removes the 2nd router's address, and pads the onion to its original length



Forwarding an Onion

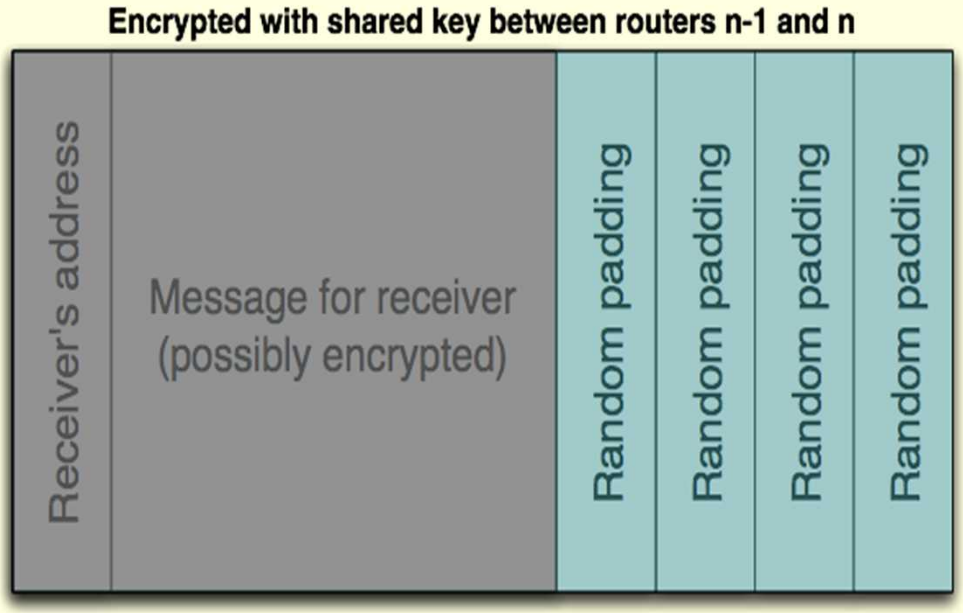
- It then encrypts the result using a symmetric key shared with the second router and forwards the result


Encrypted with shared key between routers 1 and 2
. . .
Router n-1's address
Router n's address
Receiver's address
Message for receiver (possibly encrypted)
Random padding

Forwarding an Onion

- This process is repeated at each router along the path, with each successive address stripped, and the message is padded back to its original length before re-encrypting

- Eventually the onion reaches the receiver who can extract the message



Encrypted with shared key between routers n-1 and n

Receiver's address

Message for receiver (possibly encrypted)

Random padding

Random padding

Random padding

Random padding

Onion Routers

- Due to the layered encryption, each router knows only the identity of the next and previous hops in the path
  - None of the routers even know their position in the chain
- Thus, one uncompromised router in the chain ensures sender/receiver-unlinkability
  - Note that the sender may operate a router

Reply Onions

- A sender can also construct a reply onion to allow for bidirectional communication.
  - The sender randomly selects a path through the routers, with itself as the final destination
  - It encrypts its own address with the public key of the last router in the path, prepends the last router’s address and encrypts the result with the next to last router’s public key and so on, with the final result encrypted in the receiver’s public key
- The receiver can use this to send messages back to the sender without the need to maintain logs

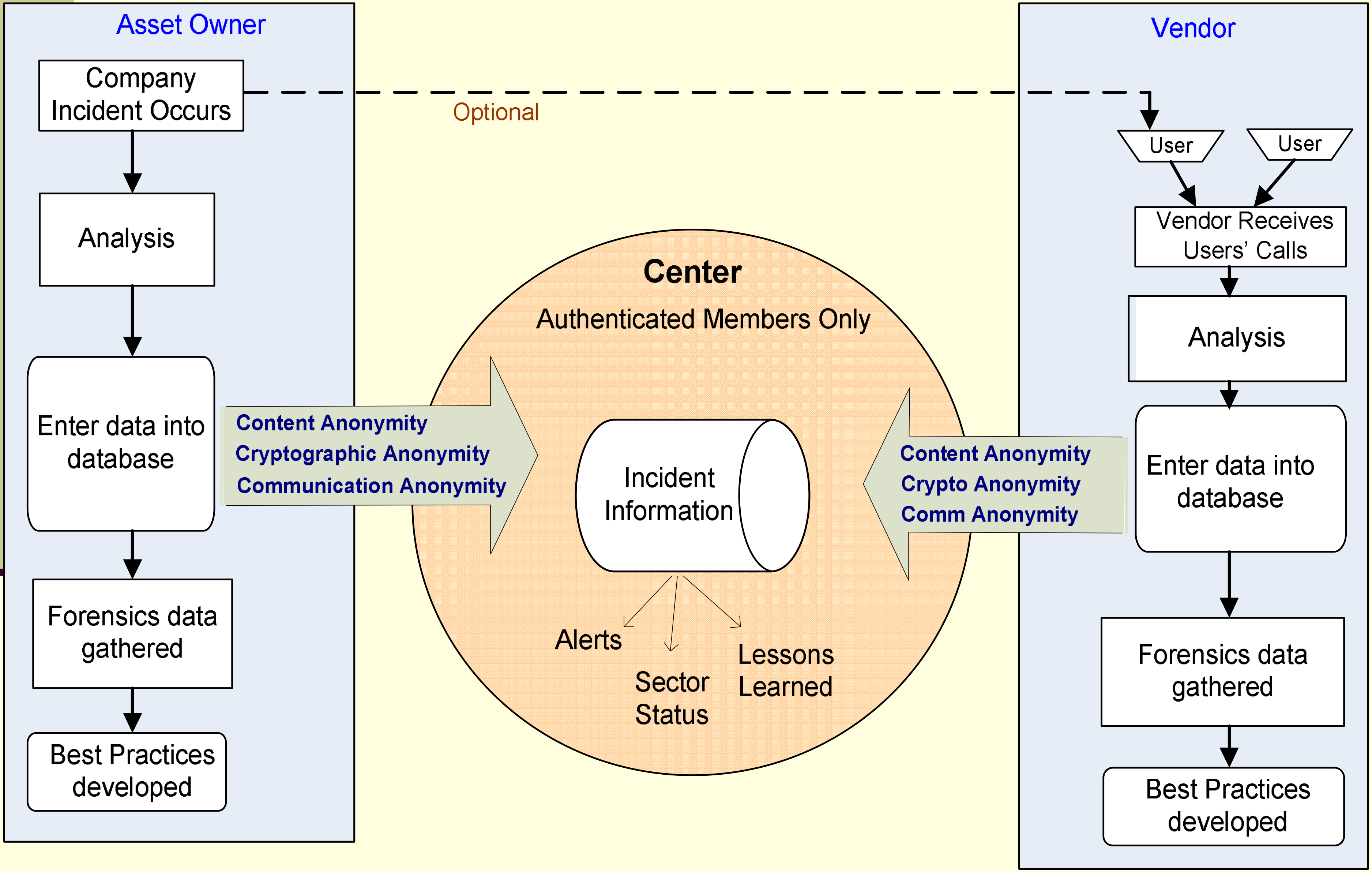
Onions Are Not Enough

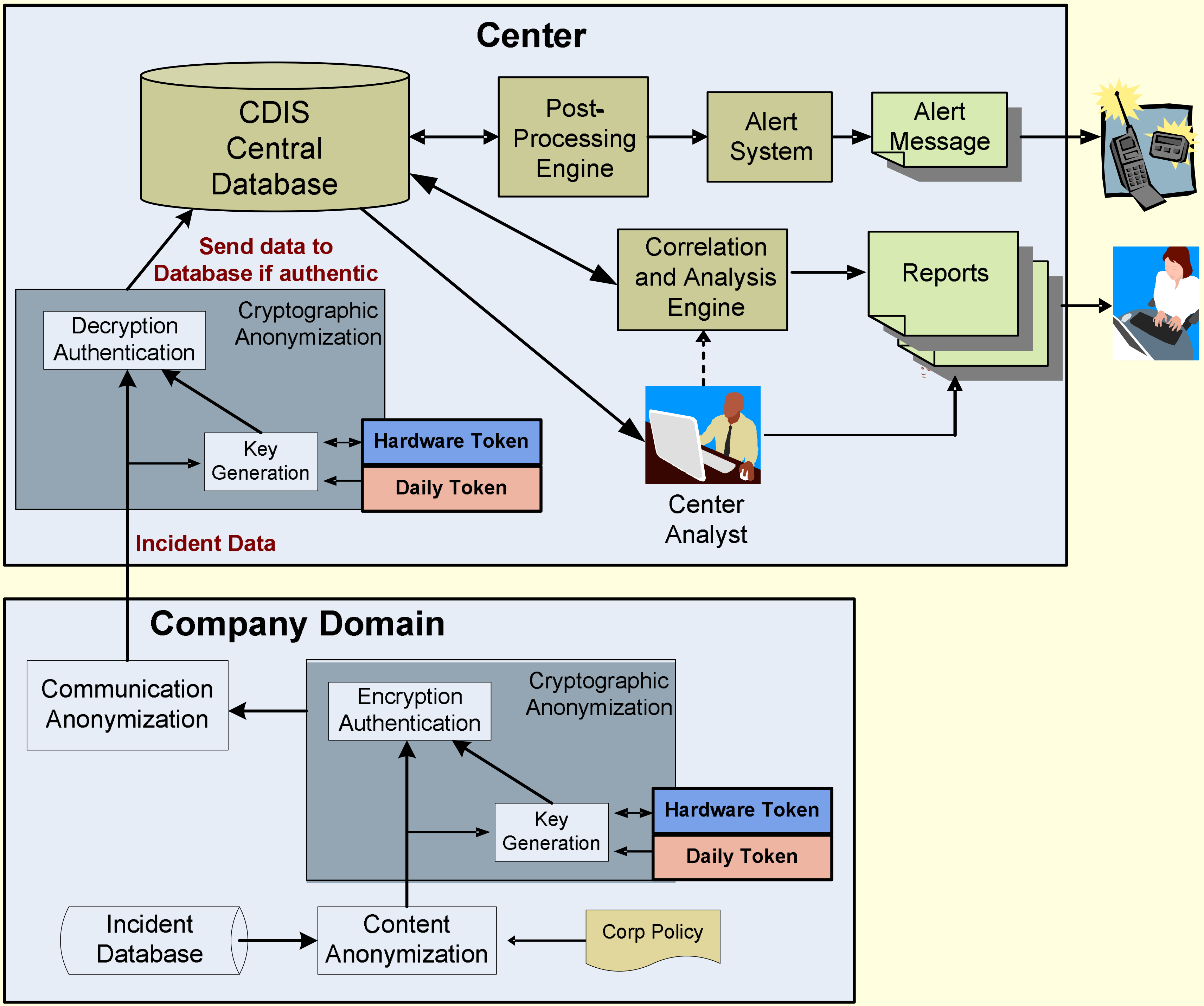
- While onion routers provide strong and flexible anonymous communications, they are not enough for our needs
- Onion routers provide no authentication
  - Anyone with knowledge of the network structure can send messages through it
  - Messages need to have authentication information attached before injection into the network
    - This is where our cryptographic anonymous authentication protocol comes into play

Content Anonymization

- A non-Cryptographic process
- Manual Content Anonymization
  - Establish Clear Policies and Operational Procedures
  - Train users on how to implement policy
- Automated Content Anonymization
  - Structured forms with no free-text
  - Information review and keyword filtering/modification
  - Bayesian filtering

SCADA / PCS Application





Center

CDIS Central Database

Post-Processing Engine

Alert System

Alert Message

Correlation and Analysis Engine

Reports

Send data to Database if authentic

Decryption Authentication

Cryptographic Anonymization

Key Generation

Hardware Token

Daily Token

Center Analyst

Incident Data

Company Domain

Communication Anonymization

Encryption Authentication

Cryptographic Anonymization

Key Generation

Hardware Token

Daily Token

Incident Database

Content Anonymization

Corp Policy

Other Applications

- Electronic voting
- Site access control
- Other critical infrastructures
  - Transportation
  - Water
  - Electric

Status

- Impact
  - Promote information sharing among group of authenticated members
  - Potential use by information sharing groups such as PCSF, ISACs, and CERT
- Accomplishments
  - Original paper published at 2001 IEEE Systems, Man, and Cybernetics Information Assurance Workshop
  - Implementation integrated into I3P CDIS Demo
  - Outreach at PCSF Workshop and I3P Workshop
- Plans
  - Demo in I3P CDIS System
  - I3P Control Systems Security Workshop
    - Houston, February 15-16, 2007
  - Technology transfer to other info sharing groups