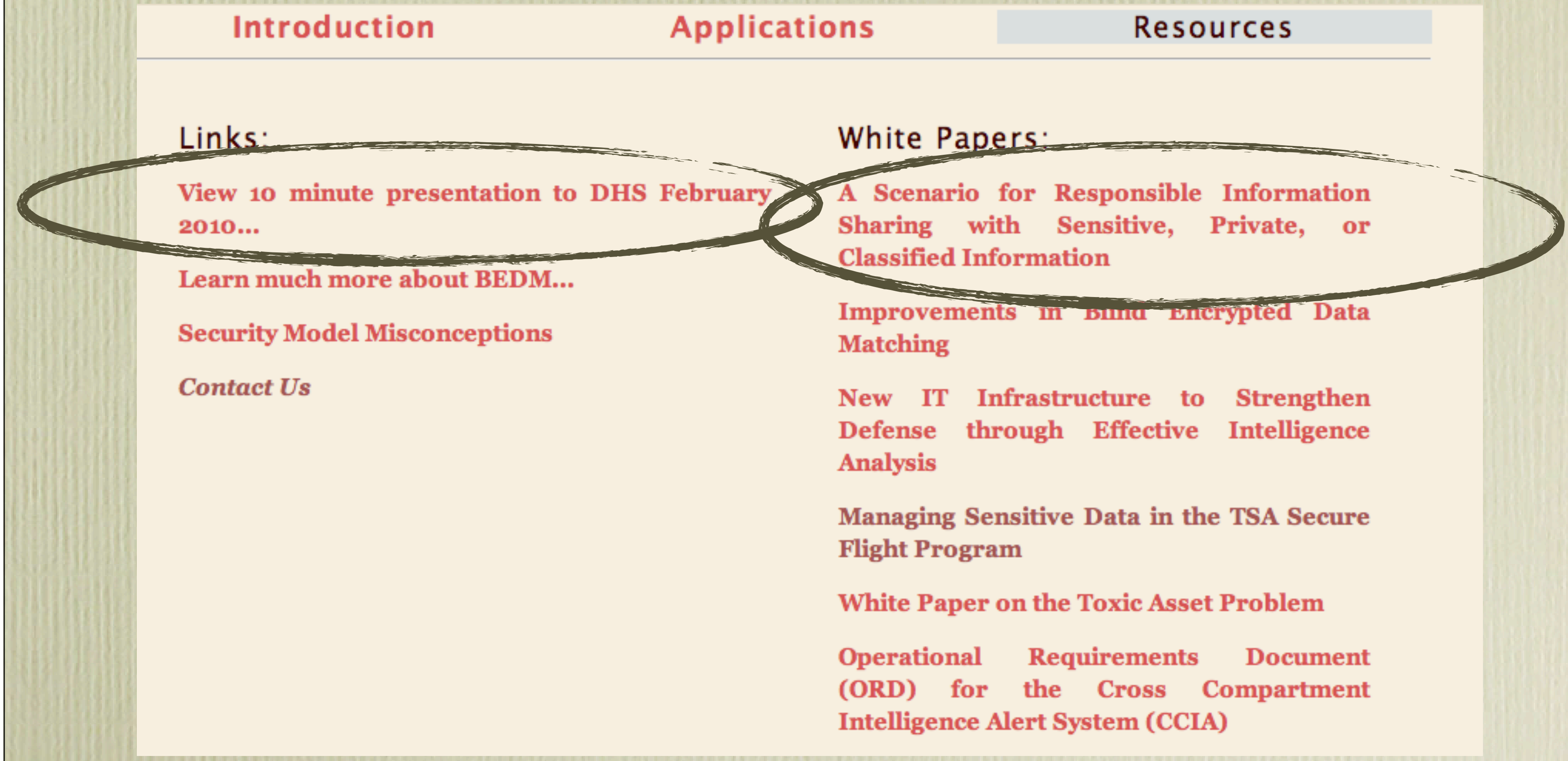
The Pygar Project

Open Source Software for BEDM: Blind Encrypted Data Matching

Paul Baker
pbaker@wwnsoftware.com

Rationale for BEDM - see: www.wwnsoftware.com/resources.html



Links:

View 10 minute presentation to DHS February 2010...

Learn much more about BEDM...

Security Model Misconceptions

Contact Us

White Papers:

A Scenario for Responsible Information Sharing with Sensitive, Private, or Classified Information

Improvements in Blind Encrypted Data Matching

New IT Infrastructure to Strengthen Defense through Effective Intelligence Analysis

Managing Sensitive Data in the TSA Secure Flight Program

White Paper on the Toxic Asset Problem

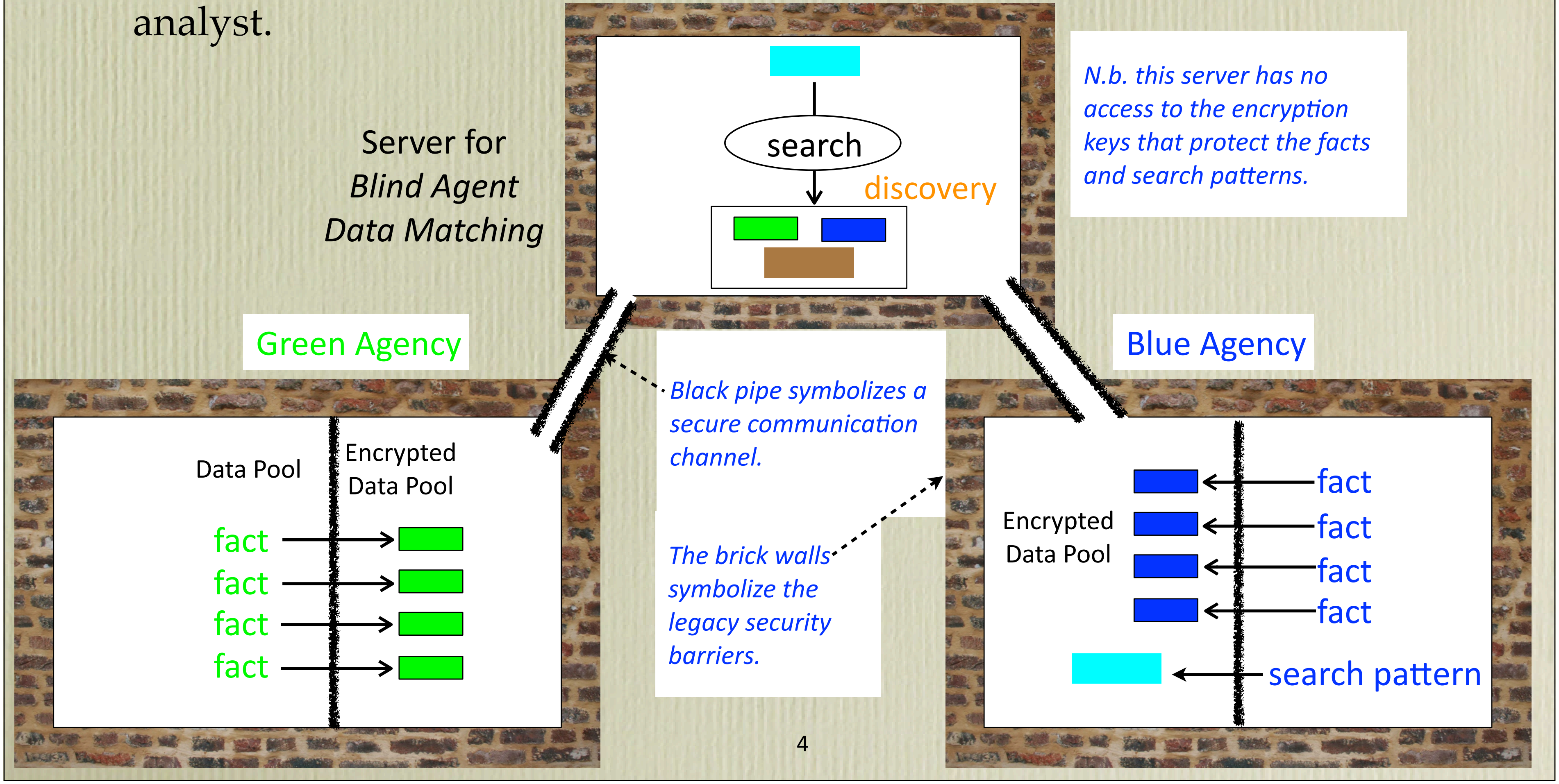
Operational Requirements Document (ORD) for the Cross Compartment Intelligence Alert System (CCIA)

Introduction

- Goal - Improve cooperation leading either to the selective sharing of secret information or to a negotiated agreement.
- Method - a blind agent facilitates the discovery and sharing of relevant information by:
  - finding connections between separately held secrets whereby a connection provides a justification to share the secrets.
  - finding a mutually agreeable set of terms from separately held, secret negotiation terms.
- Status - an open source project coordinated via SourceForge, web and blog:
  - Software components (in Java SE 6). (Pygar Project on SourceForge)
  - Interface definitions + sample implementations
  - Documentation
  - Web: http://www.wwnsoftware.com/OpenBEDM
  - Blog: http://ectn.typepad.com/pygar

Blind Encrypted Data Matching (BEDM) Heightens Security for Information Sharing - Step 1

- BEDM performs a search on encrypted facts (data) and encrypted patterns to discover facts linked by a search pattern submitted by an analyst.



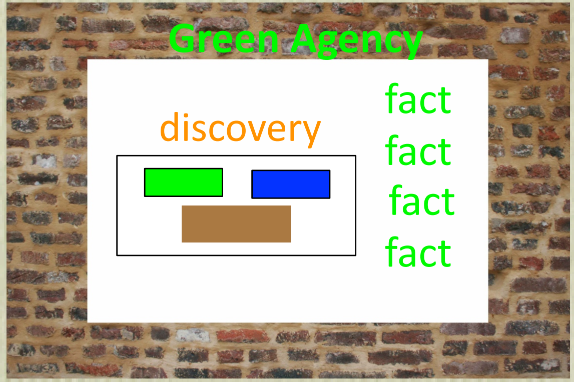
N.b. this server has no access to the encryption keys that protect the facts and search patterns.

Black pipe symbolizes a secure communication channel.

The brick walls symbolize the legacy security barriers.

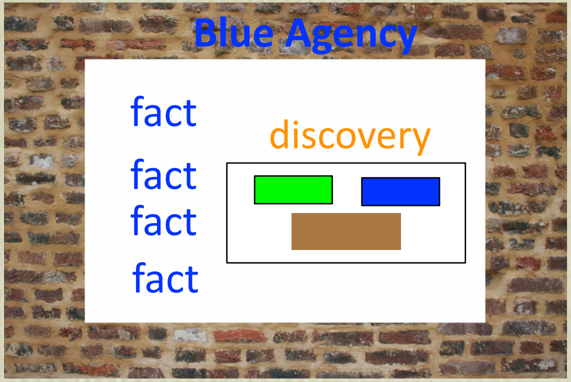
Blind Encrypted Data Matching (BEDM)
Steps 2 and 3

- Step 2: The blind-agent server shares only the discoveries - encrypted facts plus the linkage between them.

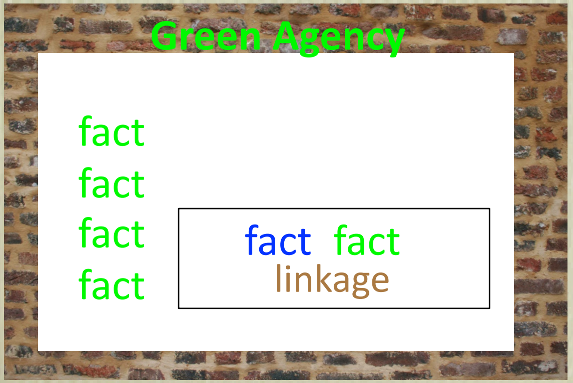


Please note, agencies can establish a different policy governing the return of encrypted discoveries.

The server will enforce the policy agreed by the agencies

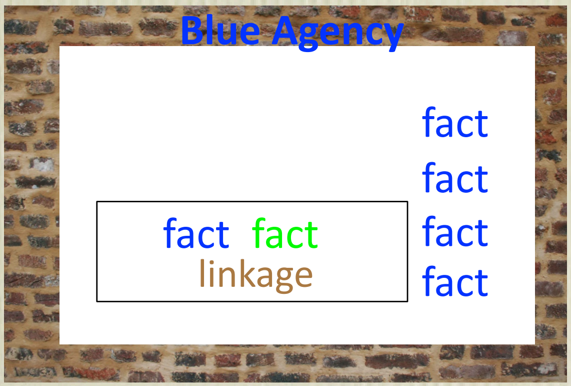


- Step 3: Green Agency and Blue Agency both have the encryption key to decrypt the analysis linkage, draw conclusions, and take action.



Both agencies see the results.

Both agencies avert security risk by maintaining encryption during the search operation.



Key Points - 1

- ❖ Security Features
  - Multiple encryption key system with at least 3 separate keys.
  - One key is a temporary session key - changes for additional security.
  - Strict isolation of the blind agent from any site where there is access to a session key.
  - Strict isolation of any client who has a session key from the consolidated encrypted data used by the blind agent for discovery and negotiation.
  - With the session key and the consolidated encrypted data strictly separated, only encrypted operations are possible and only encrypted results produced.

Key Points - 2

- Feasibility Features:
  - The session key encryption of data, queries, and parameters is a partial encryption in the sense that language tokens that define syntax are left unencrypted while any semantic information is encrypted. For example, XML tags are unencrypted, everything else is encrypted.
  - Partially encrypted data can be parsed into a grammar tree whose leaves are encrypted.
  - Subtrees of the parse trees can be compared and matched using encrypted operations on the encrypted leaves.
  - Encrypted operations are possible. There are a useful number of these operations, although there is no way to generalize from an arbitrary function to a function on encrypted data.

Encrypted Operations

- Equivalence: if x == y and if x’ = E(K, x) is an encryption function of x with a key K, then E(K, x) == E(K, y) and x’ == y’.
- Set membership: given an x and a set {x1, x2, x3 … xn} we can determine if x is a member of the set by evaluating whether the encrypted value of x, x’, is a member of the encrypted set {x1’, x2’, x3’, …xn’}. This is valuable for matching a request with a choice of options or matching one name alias with a list of name aliases.
- If P(k, x) is an order preserving encryption of x with key K, then we can determine whether x < y from the encrypted comparison x’ < y’. This is valuable for matching fuzzy terms such as biometrics or geographic location reports. (n.b. there is cause to regard the phrase “order preserving encryption” as an oxymoron).
- Using homomorphic encryption, we can find encryption algorithms for which a variety of simple numeric functions have encrypted equivalents. But the computation cost is extremely high.
- Some useful operations have no encrypted equivalent.

Why Open Source?

❖ The data involve big secrets – operations are under attack and must be guarded – but the software is open source because of Kerckhoff's principle in cryptography:

"In 1883 Auguste Kerckhoffs[1] wrote two journal articles on *La Cryptographie Militaire*,[2] in which he stated six design principles for military ciphers. Translated from French, they are:[3]

1. The system must be practically, if not mathematically, indecipherable;
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. It must be portable, and its usage and function must not require the concourse of several people;
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe. "

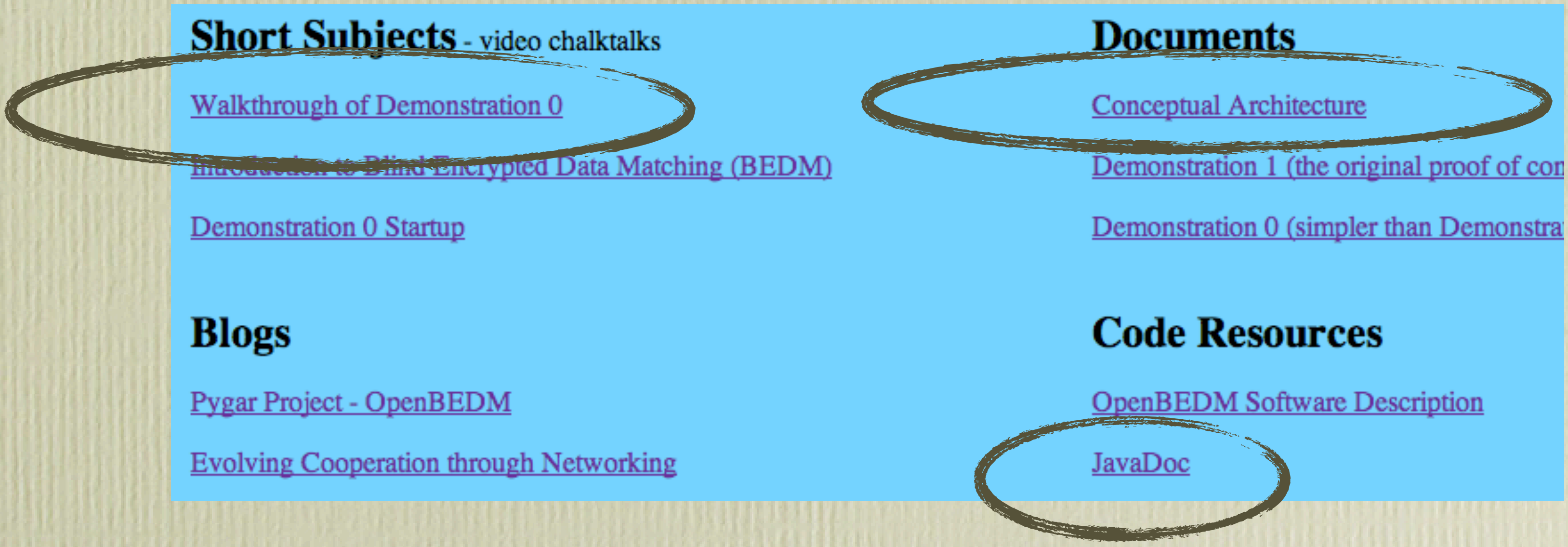
(copied from Wikipedia.Org)

Cooperation

The Pygar Project's overarching goal is the improvement of cooperation in society and in government wherever networked information systems are employed. Cooperation has been studied. Here are some recognized principles:

- Repeated peer-peer interaction: cooperation emerges in groups when the members expect to interact profitably and often. When people cooperate, altruism emerges from self-interest to maximize long-term gain. However, peers require protection from victimization and major loss in any one interaction. This is the main principle behind OpenBEDM software.
- Norms - successful groups of cooperators have norms for behavior and enforce those norms. BEDM makes enforcement of norms more difficult therefore there is an supplemental process call adjudication. OpenBEDM does not yet support adjudication.
- Metanorms - it is observed that successful groups of cooperators punish those who fail to enforce norms. This is new research topic is controversial and hard to interpret.
- Quorum sensing - recent biological research extends the empirical basis of cooperation to bacteria - while not directly relevant to software - it is interesting to observe even the simplest systems can switch from 100% self-interest to cooperation for shared benefit when they sense a critical mass. One may hope that OpenBEDM has an emergent effect beyond its relatively simple technical operation.

Other informative links on www.wwnsoftware.com/OpenBEDM



Short Subjects - video chalktalks

Walkthrough of Demonstration 0

Introduction to Blind Encrypted Data Matching (BEDM)

Demonstration 0 Startup

Blogs

Pygar Project - OpenBEDM

Evolving Cooperation through Networking

Documents

Conceptual Architecture

Demonstration 1 (the original proof of con

Demonstration 0 (simpler than Demonstra

Code Resources

OpenBEDM Software Description

JavaDoc

http://www.wwnsoftware.com/OpenBEDM

The Pygar Project

Open Source Software for BEDM:
Blind Encrypted Data Matching

Security

Sharing

WWN Software

http://www.wwnsoftware.com

Paul Baker
pbaker@wwnsoftware.com

Todays slides: http://www.wwnsoftware.com/OpenBEDM/WG4.pdf