Making Privacy-Preserving Data Mining Practical with Smartcards

Andrew Lindell

Aladdin Knowledge Systems & Bar-Ilan University

Joint work with Carmit Hazay (Bar-Ilan University)

A Real Problem

- **In many states, voters are not allowed to vote in both the Republican and Democratic primaries**
  - Thus they cannot be members of both parties
- **What can we do to enforce this law?**
  - What if we have suspicions that this behavior has become common?

Recently in Israel…

- **This problem arose in Israel’s recent primaries between the Kadima and Likud parties**
  - Kadima demanded that Likud hand over its list of party members (Kadima was spinoff from Likud)
  - Likud refused, but eventually agreed that parties hand over their membership lists to the Attorney General’s office to carry out the check
- **This is an outrageous solution**
  - Party membership is confidential (this is almost the same as revoking vote confidentiality for these citizens)

The Same Problem

- **Comparing lists of suspects**
  - If two or more agencies list the same suspect, then this calls for more investigation
- **How can we compare lists without revealing their content**
  - Of course, we wish to reveal the identities of those on both lists, but nothing else

A Different Problem

- **Can a CIA agent search the FBI database?**
  - Sometimes this is essential, but it should be limited
- **Privacy is on both sides**
  - The FBI wants/needs to limit the searches by the CIA
  - The CIA doesn't necessarily want the FBI to know what it's searching for

Why Is This Important?

- **Many different security agencies coexist**
- **These agencies are hesitant to share information**
  - This is often justified
    - If all agencies share all information, a single mole can compromise all agencies
    - “If you have one gigantic database, you have one gigantic target for the terrorists and the bad guys”, Peter Swire
- **But more patterns could be found if data and not just conclusions are shared**

In General…

- **Privacy-preserving distributed data mining**
  - **Distributed data mining/computations:**
    - Data is spread over different sites
    - Wish to compute a data mining or other algorithm on the ***union*** of the databases (increase ***UTILITY***)
  - **Privacy:**
    - We want to reveal only the outcome of the computation
    - This minimizes information flow and maximizes privacy

Another Example

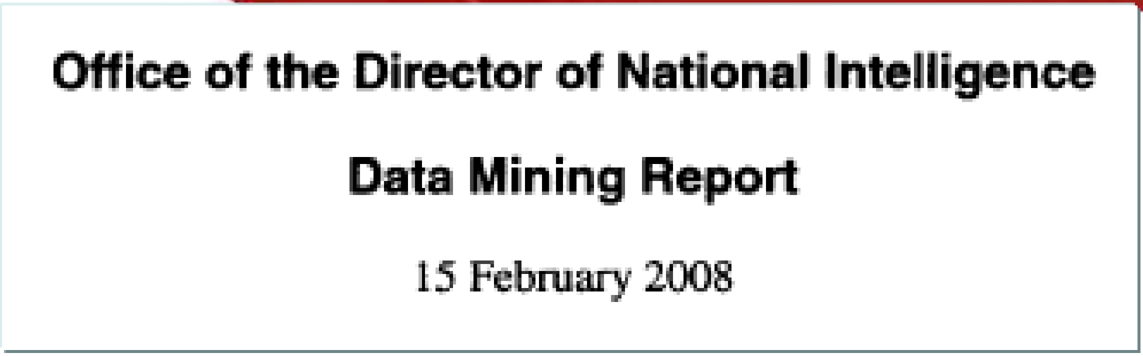
- **Investigation at Stillwater State Correctional Facility, Minnesota**
  - Data mining software was applied to phone records from the prison
  - A pattern linking calls between prisoners and a recent parolee was discovered
  - The calling data was then mined again together with records of prisoners’ financial accounts
  - The result: a large drug smuggling ring was uncovered
- **What about the privacy concerns?**

Secure Computation

- **Compute a function of private inputs held by different parties so that**
  - **Privacy:** no party learns anything beyond the output
  - **Correctness:** the output is guaranteed to be correct
  - **Independence of inputs:** one party cannot make its input depend on other parties' inputs
- **Security must be preserved in the presence of adversarial behavior**
  - **Semi-honest adversaries:** follow protocol but try to learn more from transcript
  - **Malicious adversaries:** follow arbitrary attack strategy

Efficient Secure Computation

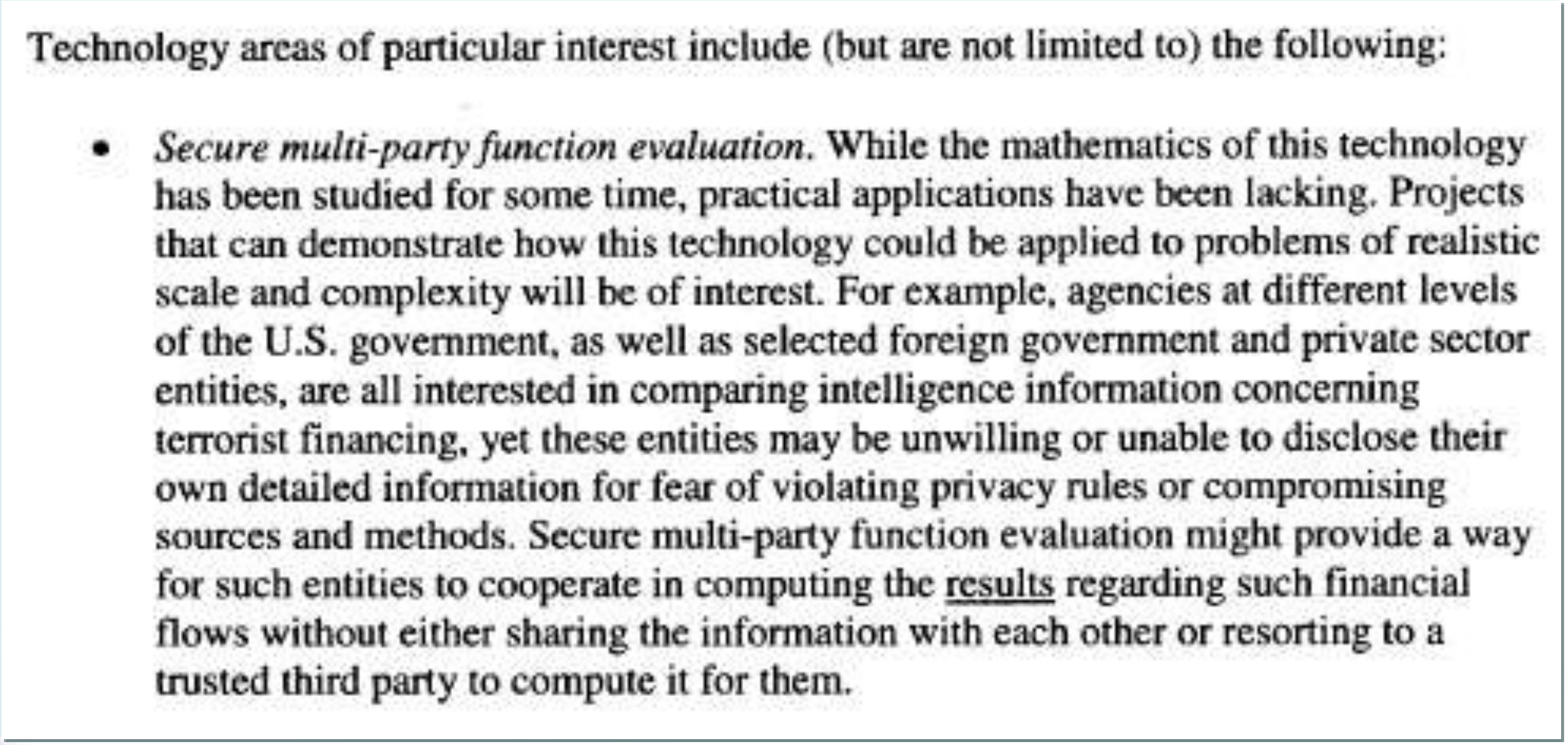
- **Extremely hard to achieve!**
- **In the semi-honest adversary model**
  - We have a large number of reasonably efficient protocols (but even here, they typically require something like an exponentiation per input bit)
  - But the semi-honest model is very weak
    - It is appropriate for preventing inadvertent leakage but not much more
- **In the malicious adversary model**
  - Few highly efficient protocols exist



Office of the Director of National Intelligence

Data Mining Report

15 February 2008



Technology areas of particular interest include (but are not limited to) the following:

- *Secure multi-party function evaluation.* While the mathematics of this technology has been studied for some time, practical applications have been lacking. Projects that can demonstrate how this technology could be applied to problems of realistic scale and complexity will be of interest. For example, agencies at different levels of the U.S. government, as well as selected foreign government and private sector entities, are all interested in comparing intelligence information concerning terrorist financing, yet these entities may be unwilling or unable to disclose their own detailed information for fear of violating privacy rules or compromising sources and methods. Secure multi-party function evaluation might provide a way for such entities to cooperate in computing the results regarding such financial flows without either sharing the information with each other or resorting to a trusted third party to compute it for them.

Bridging Theory and Practice

- **Approach 1**
  - Come up with weaker definitions of security that are still strong enough
    - This has potential, but still difficult (even hard to get very high efficiency for semi-honest adversaries)
- **Approach 2**
  - Change our assumptions regarding the resources parties have to carry out their computations
- **We follow approach two in our work here**

Our Results

- **We present simple and truly practical protocols, that use standard smartcards and standard smartcard infrastructure**
    - With security in the presence of malicious adversaries

What is a Smartcard?

- **A smartcard is a secured piece of hardware with well-defined functionality**
- **Smartcards store cryptographic keys and can carry out operations on-board**
  - The keys never leave the smartcard
- **Smartcards have strong physical protection**
  - Self-destruct if exposed to light, or if triggered
  - Obfuscated logic
  - Miniaturization to make reverse engineering hard
  - And much much more…

Are Smartcards Unbreakable?

- **No!**
  - But high-end certified smartcards are very hard to break (requiring great expertise, time and expensive equipment)
- **Is it acceptable to assume that the smartcards we use are not breakable?**
  - It depends on the application
  - We personally find it a more reasonable assumption than the assumption that the code a user is running is not corrupted…

Smartcard Aided Computation

- **The computation is carried out by the parties communicating over a network**
- **In addition, at some stage one party prepares a standard smartcard (in some way) and physically sends it to the other**
  - The same smartcard can be reused many times
    - Also for different protocols
- **This model is suitable for non-transient applications (e.g., homeland security or interaction between government agencies)**

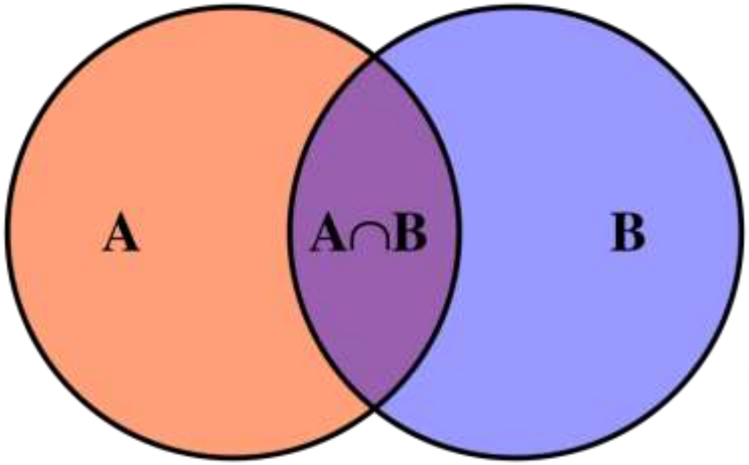
Why Standard Smartcards?

- **Trust**
  - Can buy smartcards from a third-party vendor with no personal interest (and with a lot to lose)
- **Ease of deployment**
  - Can use any smartcard off the shelf
  - Note: smartcards are becoming more and more ubiquitous
    - Smartcard logon
    - Digital signatures
    - Laptop protection

Protocol 1– Set Intersection

- **Set Intersection**
  - **Input**: two or more parties with private databases (keyed by some attribute, say SSN)
  - **Output**: the keys that appear in both databases (e.g., social security numbers appearing in both), and **nothing more**

- **Many applications…**



Andrew Lindell
Aladdin Knowledge Systems

Preliminaries

- **Pseudorandom functions**
  - A random function is a function that assigns a random output to every input (independently of all others)
  - A pseudorandom function is a cryptographic function that looks like a random one
    - It uses a secret key and is efficiently computable (given the key)

Pseudorandom Functions

- **Modern block ciphers are constructed to be pseudorandom functions**
  - 3DES and AES
  - From the AES call for candidates – “algorithms will be judged on the following factors. . .
    - The extent to which the algorithm output is indistinguishable from a random permutation on the input block”
- **Denote a pseudorandom function with secret key k by E_k**

The Protocol – Basic Idea

- **Input: a set $X = \{x_1,\ldots,x_n\}$ held by Alice, and a set $Y = \{y_1,\ldots,y_n\}$ held by Bob**
- **Protocol :**
  - Alice chooses a secret key **k** and imports it into a smartcard that is sent to Bob
  - In addition, Alice computes $\mathbf{X_E} = \{\mathbf{E_k(x_1)},\ldots,\mathbf{E_k(x_n)}\}$ and sends the set $\mathbf{X_E}$ to Bob
  - Bob computes the values $\mathbf{E_k(y_1)},\ldots,\mathbf{E_k(y_n)}$ obliviously using the smartcard
  - Bob outputs the set of values $\mathbf{y_i}$ for which $\mathbf{E_k(y_i)} \in \mathbf{X_E}$

The Protocol Idea – Graphically

Alice

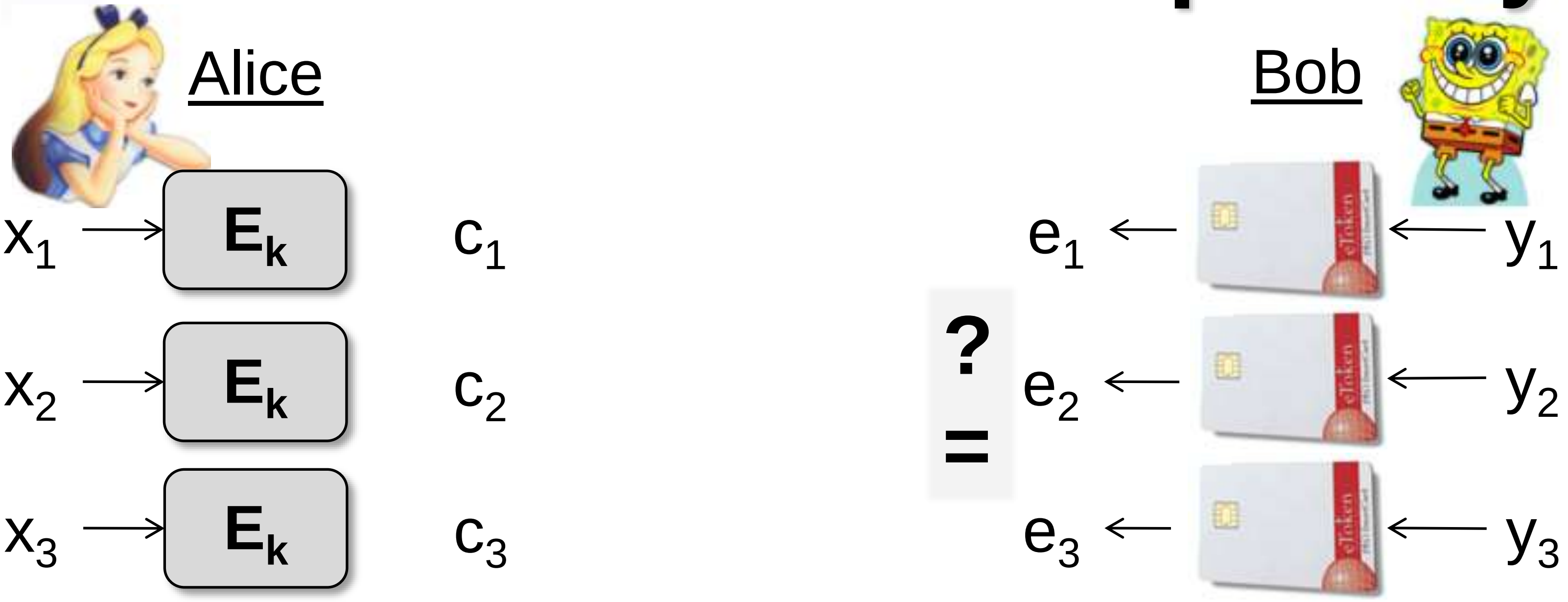
Bob

$x_1 \rightarrow \boxed{E_k} \rightarrow c_1$ y_1

$x_2 \rightarrow \boxed{E_k} \rightarrow c_2$ y_2

$x_3 \rightarrow \boxed{E_k} \rightarrow c_3$ y_3

The Protocol Idea – Graphically



Alice

Bob

$x_1 \rightarrow \boxed{E_k}$ c_1

$x_2 \rightarrow \boxed{E_k}$ c_2

$x_3 \rightarrow \boxed{E_k}$ c_3

$\overset{?}{=}$

$e_1 \leftarrow$ $\leftarrow y_1$

$e_2 \leftarrow$ $\leftarrow y_2$

$e_3 \leftarrow$ $\leftarrow y_3$

- **Bob compares e_1,e_2,e_3 to all values c_1,c_2,c_3**
  - Any value $\mathbf{e_i}$ appearing in the set {$\mathbf{c_1}$,$\mathbf{c_2}$,$\mathbf{c_3}$} is in the intersection
    - This is because the smartcard computes $\mathbf{E_k}$ for the same **k**
  - For example, if $\mathbf{e_1}$=$\mathbf{c_2}$, then $\mathbf{y_1}$=$\mathbf{x_2}$ and so $\mathbf{y_1}$ is in the intersection

The Security – Basic Idea

- **Alice doesn't learn anything from the protocol**
  - Alice doesn't receive any message from Bob so this is trivial

- **Bob learns only what is in the intersection, and nothing else**
  - This follows from the fact that a value $\mathbf{E_k(x)}$ reveals nothing about **x** unless Bob explicitly queries the smartcard with **x**
    - This is due to the presumed security of **E**

Achieving Provable Security

- **For technical reasons, a minor modification is needed to obtain a rigorous proof of security**
- **Protocol :**
  - Alice chooses a secret key **k** and imports it into a smartcard that is sent to Bob
  - Bob computes the values $\mathbf{E_k(y_1)},\ldots,\mathbf{E_k(y_n)}$ obliviously using the smartcard and announces to Alice that he has finished
  - Alice erases the key **k** from the smartcard
  - Alice computes $\mathbf{X_F} = \{\mathbf{E_k(x_1)},\ldots,\mathbf{E_k(x_n)}\}$ and sends the set $\mathbf{X_F}$ to Bob
  - Bob outputs the set of values $\mathbf{y_i}$ for which $\mathbf{E_k(y_i)} \in \mathbf{X_F}$

Alice and the Smartcard

- **Alice needs to**
  - Import a key **k** to the smartcard
  - Erase the key (and ensure that it was indeed erased)
- **How is this achieved without physically sending the card back and forth?**
- **Secure messaging**
  - Alice creates a directory on the smartcard such that importing a key to the directory and erasing from it is carried out using encryption and message authentication
  - Only Alice knows the secret keys for encrypting and authenticating

Secure Messaging – More Details

- **Secure messaging – encryption:**
  - Alice shares a secret key with the smartcard (associated with some directory)
  - All messages related to that directory (e.g., import key, etc.) are encrypted with that key
- **Secure messaging – authentication:**
  - Alice shares another secret key with the smartcard
  - All messages related to that directory are MACed with that key; this prevents Bob from modifying any message
  - These messages include **replies** from the smartcard authorizing that an operation succeeded

Alice and the Smartcard

- **Alice imports a key to the smartcard**
  - The key is encrypted and so Bob knows nothing about it
- **Alice erases the key from the smartcard**
  - The return value from the smartcard, authorizing that the erasure succeeded, is MACed
  - This means that Bob cannot forge such an authorization and so must erase the key, as specified

Reusing the Smartcard

- Since secure messaging is deployed, a new key can be imported whenever the protocol needs to be run
- This means that a smartcard can be sent once from each party to the other, and then can be reused many times (for this protocol and for others)

A Subtle Point

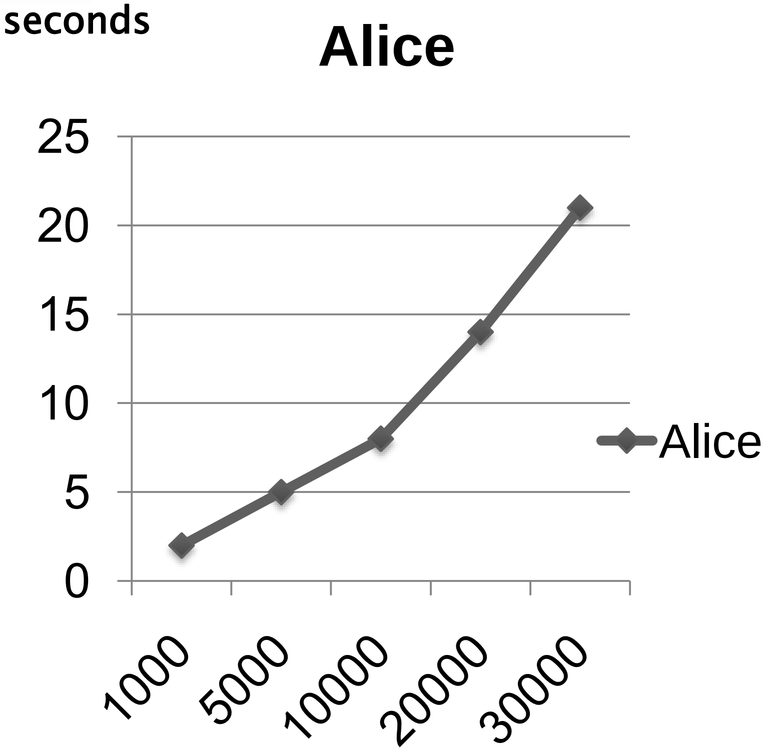
- **What prevents Bob from querying the smartcard on a huge number of values (to run an exhaustive search)?**
  - Smartcard objects can be initialized with a “usage counter” limiting the number of times an object can be used
  - When Alice initializes the smartcard with a key for **E**, she sets the usage counter to equal the size of Bob’s input set
  - **Note**:
    - Bob can always lie about the size of his set, but not by too much (or Alice will become suspicious)
    - Other means can also be used to prevent this (authorization from other sources regarding the size of the set)

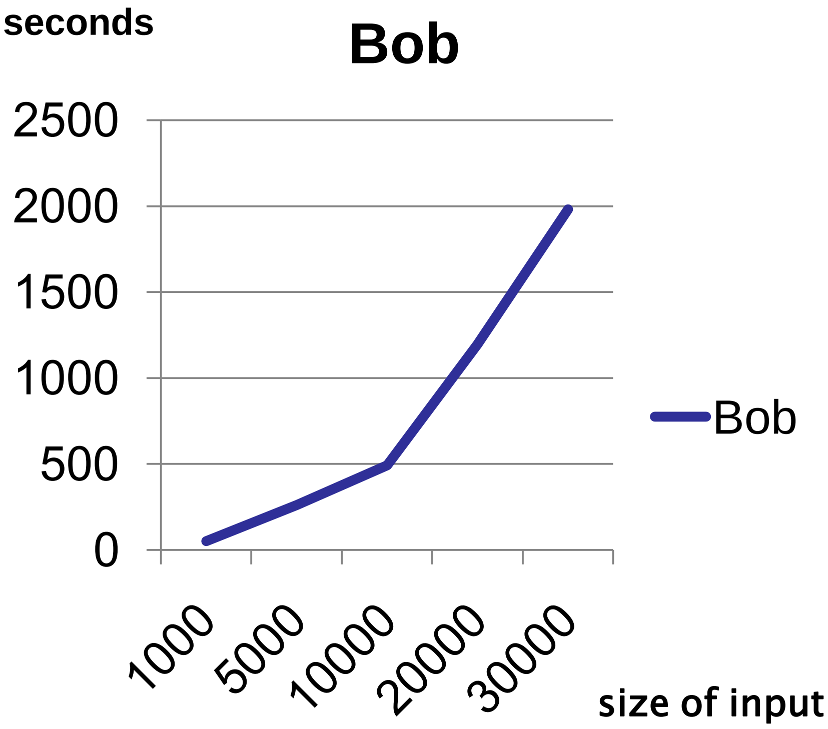
A Demo

- **We implemented the protocol using Aladdin's eToken PRO**
  - No attempt has been made to optimize the code
  - Nevertheless, it is very efficient
  - For 10,000 records (using an IBM T41p laptop)
    - Alice: a few seconds
    - Bob: 9 minutes (and can be parallelized)

* Thanks to Danny Tabak of Aladdin for the implementation!

Experimental Results





Properties of the Protocol

- **Highly efficient**
  - Alice carries out all pseudorandom function operations on her PC
  - Bob computes one smartcard operation per input value
- **Provable security**
  - The protocol can be proven secure under stringent definitions, demonstrating that nothing beyond the set intersection itself can be learned
- **Simple to implement**

What Else?

- **What else can be done in this model?**
- **Oblivious database search**
  - A client carries out a search on a database (retrieving a **single record** via a keyword)
  - The server learns **nothing** about what the client searched for

Oblivious DB Search

- **A trivial solution?**
  - The client downloads all of the database
- **Limiting information flow**
  - The aim of the solution is to limit the amount of information that the client obtains
  - The client is only allowed to carry out one search (or another predetermined number of searches)

A Paradox

- **How is it possible to limit the information flow without the server knowing what the client is searching?**
  - If the server knows, then it could just send the requested record
  - If the server doesn’t know, how can we limit the number of searches the client carries out?

Motivation

- **Classified databases**
  - One homeland security agency wishes to search for a suspect in a different agency's database
  - Allowing full access is dangerous
  - The identity of the suspect is also highly classified and so revealing it to the other agency is unacceptable

Commercial Application

- **LexisNexis is a search engine for legal professionals**
  - Can search for case summaries etc.
- **There are a number of payment options: one of them is pay per search**
- **Such searches can be HIGHLY CONFIDENTIAL**
  - An efficient solution to the above problem is highly desirable

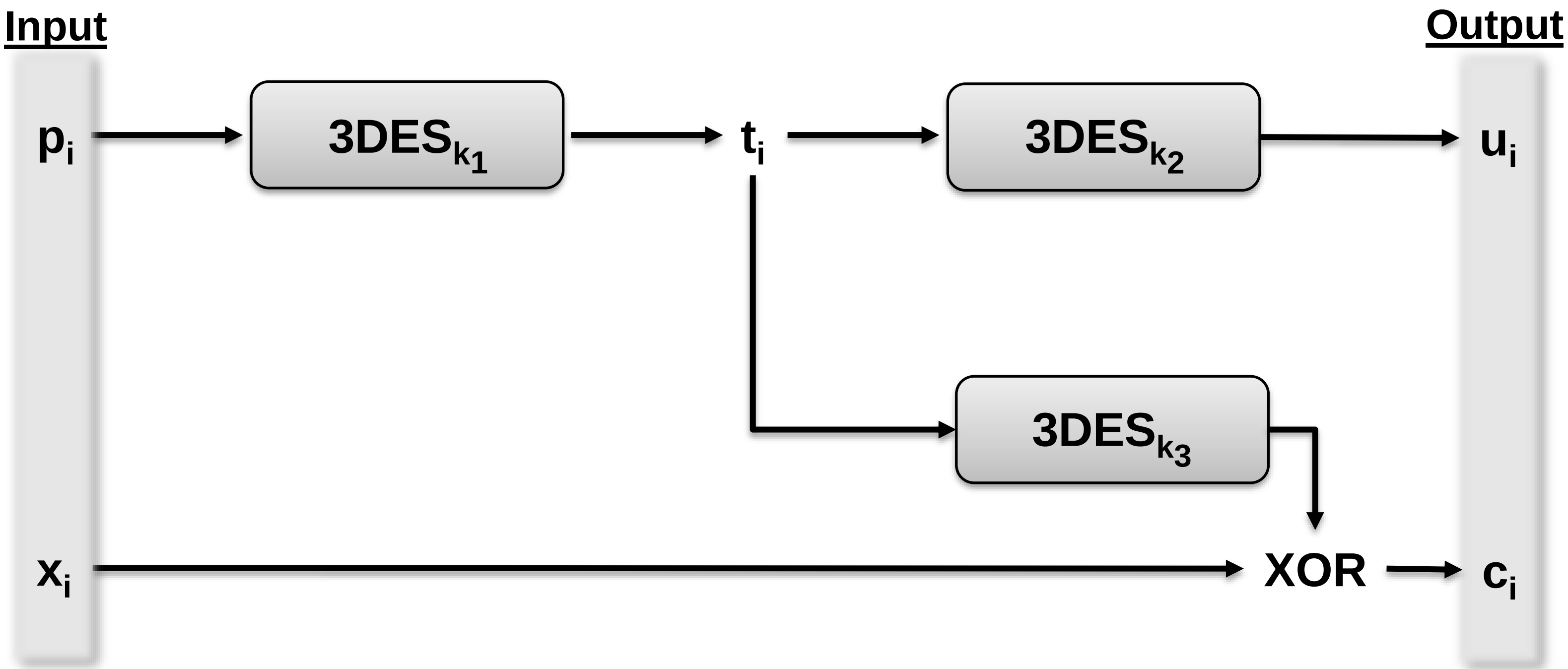
Our Solution

- **We present a solution where**
  - The server encrypts the database (in a special way) using symmetric encryption only, and one pass
  - The client downloads the database (but cannot decrypt it)
    - This download takes place only once (and updates are "pushed" to the client when necessary)
  - Each search requires a very short interaction between the client and server
- **We also present a generalization to document search by keywords**

A Solution

- **Database structure**
  - Every record contains a keyword **p** (search attribute) and a record **x**
    - The **i**th record is denoted **(p_i,x_i)**
  - The keyword **p_i** is unique in the database
- **Encrypting the database (using 3 keys k_1,k_2,k_3)**
  - Compute **t_i** = **$E_{k_1}(p_i)$** and **u_i** = **$E_{k_2}(t_i)$** and **c_i** = **$E_{k_3}(t_i)$ XOR x_i**
    - **u_i** is the new keyword value
    - **t_i** is used to mask the record value **x_i**

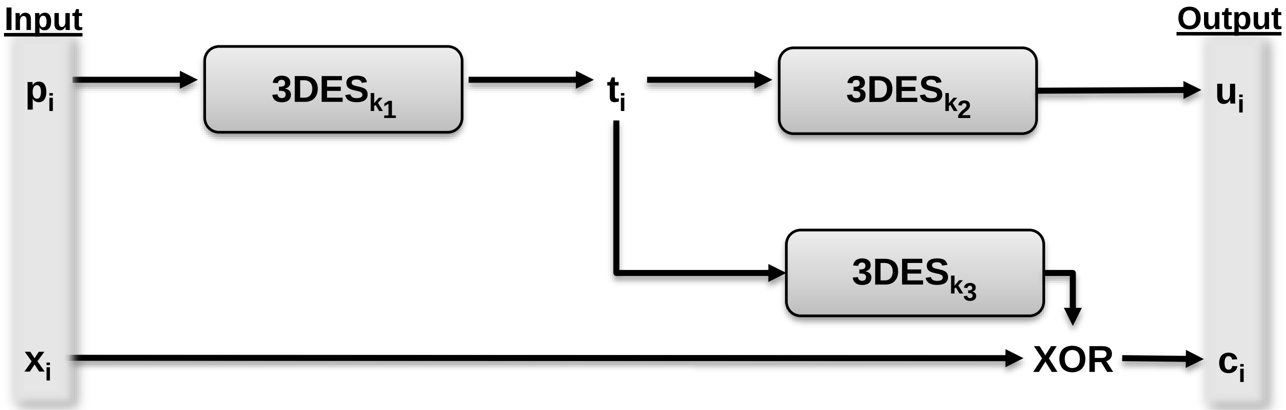
Database Encryption



Andrew Lindell
Aladdin Knowledge Systems

Database Encryption – Notes

- **Privacy: (u_i,c_i)** completely hides **(p_i,x_i)**
- **Search:** given a smartcard to compute 3DES with the 3 keys, it is possible to search
  - Compute **u**=**3DES**$_{k_2}$(**3DES**$_{k_1}$(**p**)) and see if such a **u**$_i$ exists
  - If yes, compute **3DES**$_{k_3}$(**3DES**$_{k_1}$(**p**)) **XOR c**$_i$ to obtain **x**$_i$



Aladdin Knowledge Systems

The Protocol

- **The server sends the client pairs $(u_1,c_1),(u_2,c_2),\ldots$**
- **The server sends a smartcard to the client with the keys k_1,k_2,k_3 inside**
  - The usage counter is set to the number of searches allowed to the client
- **With keyword p, the client computes $t = 3DES_{k_1}(p)$ and $u = 3DES_{k_2}(t)$ using the smartcard**
  - If there exists an **i** for which **u** = $\mathbf{u_i}$, then **p** is the **i**th keyword
  - Compute $\mathbf{x_i = 3DES_{k_3}(u)\ XOR\ c_i}$

The Protocol

Server

Client

Choose random $\mathbf{k_1,k_2,k_3}$
Initialize smartcard →

For every $\mathbf{i} = \mathbf{1},\ldots,\mathbf{n}$ compute:

- $t_i = \mathbf{3DES}_{k_1}(p_i)$
- $u_i = \mathbf{3DES}_{k_2}(t_i)$
- $c_i = \mathbf{3DES}_{k_3}(t_i)$ **XOR** x_i

$(u_1,c_1),\ldots,(u_n,c_n)$ →

Let **p** be keyword to search

p →
t ←
t →
u ←
t →
mask ←

If $\mathbf{u_i}$ exists, retrieve $\mathbf{c_i}$ and output $\mathbf{x = c_i}$ **XOR mask**

Security Analysis

- **The server cannot learn anything**
  - It only sends information
- **The client learns only the predetermined number of queries**
  - Without explicitly searching for some $\mathbf{p}=\mathbf{p}_i$, it is impossible to learn $\mathbf{u}_i$
  - This means that the client cannot know if $\mathbf{p}_i$ is in the database (it doesn't know the associated $\mathbf{u}_i$)
  - Furthermore, the client cannot learn $\mathbf{x}_i$ (again, without $\mathbf{u}_i$ it cannot query the smartcard to learn the mask)

Efficiency

- **The server prepares all encryptions on a regular computer (e.g., PC)**
  - Thus, the cost to the server is just that of symmetrically encrypting the database (essentially zero cost)
- **For every search, the user needs to make a *constant* number of queries to the smartcard**
  - This is of negligible cost (about 50ms per query)

A Problem

- **How can we reuse the smartcard here to allow for many searches at different times?**
  - Recall, the usage counter was set to the number of allowed searches
  - But in the general case, we may allow some today and some next week, and it may depend…
- **The solution – background**
  - **Access-granted counter**: The 3DES computation can be limited to once for every time a **test** is passed
  - The test can be a **challenge/response** using a strong cryptographic key

Reusing the Smart Card

- **The server sends the encrypted database and smartcard to the client**
- **When the client wishes to carry out a search**
  - The client requests a challenge from the smartcard
  - The server provides the response
  - The client can then carry out one search (as required)

The Full Protocol

Server

Client

Choose random $\mathbf{k_1,k_2,k_3}$
Initialize smartcard

For every **i** = **1**,…,**n** compute:

- $\mathbf{t_i} = \mathbf{3DES_{k_1}(p_i)}$
- $\mathbf{u_i} = \mathbf{3DES_{k_2}(t_i)}$
- $\mathbf{c_i} = \mathbf{3DES_{k_3}(t_i)}$ **XOR** $\mathbf{x_i}$

$(\mathbf{u_1,c_1}),\ldots,(\mathbf{u_n,c_n})$

Let **p** be keyword to search
Get challenge

challenge

Compute **response**

response

p
t
t
u
t
mask

If $\mathbf{u_i}$ exists, retrieve $\mathbf{c_i}$ and output $\mathbf{x = c_i}$ **XOR mask**

Efficiency/Usability

- **Database preparation is just symmetric encryption**
- **Answering queries is no more work than answering a regular web query (in fact, even less)**
- **The only drawback**
  - The user needs to store the entire encrypted database locally
  - Again, this is feasible for organizations (which is the target market in any case); it’s also actually done

Oblivious Document Search

- **What about the more general case of document search by keywords?**
- **This can be solved using the previous solution, as follows:**
  - Encrypt each document under a different key
  - For every keyword, define the “data” for this keyword to be the set of keys and document identifiers containing the keyword
  - Use the previous solution on this database

The Debate on Privacy

- **The debate on privacy typically offers us two alternatives**
  - Homeland security at the expense of your privacy
  - Personal privacy at the expense of your personal safety
- **If these are your choices, then there isn’t much of a choice**
  - “We reject as false the choice between our safety and our ideals.” President Barack Obama, January 2009

A Third Alternative

- **Develop technological tools for achieving personal privacy while still enabling data mining for homeland security (or anything else)**
  - At least, **maximize** personal privacy to the utmost possible

- **We argue that privacy-preserving technologies can enable (rather than hinder) information flow**
  - This is because privacy advocates will not fight to close programs…

Terrorist Information Awareness

Congressional Record: July 14, 2003 (Senate)

Page S9339-S9354

DEPARTMENT OF DEFENSE APPROPRIATIONS ACT, 2004

SA 1217. Mr. STEVENS proposed an amendment to the bill H.R. 2658, making appropriations for the Department of Defense for the fiscal year ending September 30, 2004, and for other purposes; as follows:

[...]

Sec. 8120.

(a) Limitation on Use of Funds for Research and Development on Terrorism Information Awareness Program.--Notwithstanding any other provision of law, no funds appropriated or otherwise made available to the Department of Defense, whether to an element of the Defense Advanced Research Projects Agency or any other element, or to any other department, agency, or element of the Federal Government, may be obligated or expended on research and development on the Terrorism Information Awareness program.

(b) Limitation on Deployment of Terrorism Information Awareness Program.--(1) Notwithstanding any other provision of law, if and when research and development on the Terrorism Information Awareness program, or any component of such program, permits the deployment or implementation of such program or component, no department, agency, or element of the Federal Government may deploy or implement such program or component, or transfer such program or component to another department, agency, or element of the Federal Government, until the Secretary of Defense--(A) notifies Congress of that development, including a specific and detailed description of-- (i) each element of such program or component intended to be deployed or implemented; and

[...]

(1) **the Terrorism Information Awareness program should not be used to develop technologies for use in conducting intelligence activities or law enforcement activities against United States persons without appropriate consultation with Congress or without clear adherence to principles to protect civil liberties and privacy**;

The Big Brother Database

OTTAWA, ONTARIO - The Minister of Human Resources Development Canada, the Honourable Jane Stewart, announced today that following discussions with the Privacy Commissioner, HRDC's information databank for labour market and social programs, the Longitudinal Labour Force File (LLFF), is being dismantled.

With the dismantling of the LLFF, HRDC has eliminated the computer program used to link its information with information from the Canada Customs and Revenue Agency and data on social assistance from provincial/territorial governments.

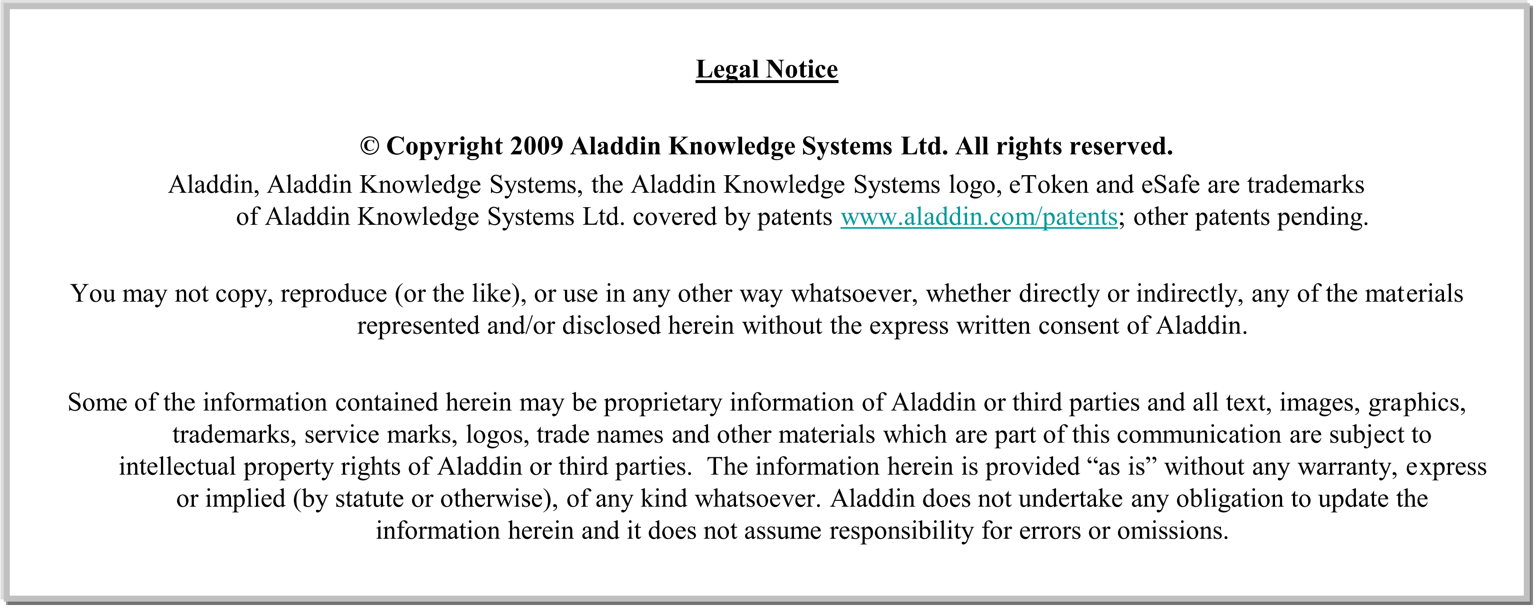
LLFF information from the Canada Customs and Revenue Agency has been returned to that Agency. HRDC will review the information-sharing arrangements it has with provincial and territorial governments for research purposes. The Department's policy analysis and research data relating to its own programs will be kept as separate, secure and unlinked files; all personal information identifying individuals will remain encrypted.

"The Privacy Commissioner fully supports this decision, and the other measures we are taking to protect privacy," said Minister Stewart. "In a letter to my department Mr. Phillips has said that he accepts and supports these measures, and that they satisfy all the recommendations and observations outlined in his 1999-2000 Annual Report."

"The Privacy Commissioner acknowledges that there has never been a known breach of security with regard to this databank, and HRDC has been acting within the existing Privacy Act. However, given public concerns about privacy issues in this era of advanced and constantly changing technology, I have chosen an approach that addresses future threats to privacy."

Summary

- **It is possible to construct secure protocols that:**
  - Have full proofs of security
  - Are efficient enough to be used in practice
  - Use standard infrastructure that exists today
- **In order to achieve this, we use smartcards**
  - We use existing infrastructure and standard, off the shelf, smartcards
- **We believe that smartcard infrastructure can be used to bridge the gap between theory and practice for secure computation**



Legal Notice