DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

DSTO

Secure Database Data Transfer with Starlight

Jyotsna Das, Brenton Williams and Greg Chase

DSTO-TR-0798

Secure Database Data Transfer with Starlight

Jyotsna Das, Brenton Williams and Greg Chase

Information Technology Division
Electronics and Surveillance Research Laboratory

DSTO-TR-0798

ABSTRACT

There is a requirement in defence to connect secure computing facilities to unclassified information sources such as the Internet. This is particularly so in the database arena where data stored in differently classified databases must be shared. The Starlight Interactive Link, was developed to connect computers of differing security classifications in a trusted manner to enable a one-way flow of data from the lower to the higher classified system. Using the Starlight Interactive Link we have developed methodologies for secure data transfer between databases resident on computing systems classified at different levels, and constructed a prototype demonstration system using these techniques.

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

Published by

DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury South Australia 5108 Australia

Telephone: (08) 8259 5555
Fax: (08) 8259 6567

AR-010-880
March 1999

APPROVED FOR PUBLIC RELEASE

Secure Database Data Transfer with Starlight

Executive Summary

Defence agencies have a requirement to integrate information stored on computers classified at different levels. The Starlight Interactive Link was developed to address the problem of connecting computers of differing security classifications in a trusted manner to enable a one way flow of data from the lower to the higher classified computer. Using this technology, we have developed methodologies for the trusted transfer of data between databases resident on computers classified at different levels.

Two techniques were developed for data transfer – one to facilitate *ad hoc* data requests and the other for pre-established routine data transfers.

The *ad hoc* method is useful for interactive querying of the database classified at the lower level on an 'on-the-fly' basis. This method does require the user to process and integrate the query results into the database at the higher end.

The pre-established routine transfers are based on the concept of establishing contracts between the two sites. Contracts specify what information is to be extracted from the low end database and how it is to be integrated into the high end database. Once the contracts are established, data transfer and integration proceeds without further user involvement. This method is particularly suited to regular querying of the low end database or database replication.

A demonstration system was implemented using these methods.

Authors

Jyotsna Das
Information Technology Division

Jyotsna is a Senior Professional Officer C in the Information Management and Fusion Group. She has worked on heterogenous database issues, database design and interfacing databases with software developed for command and control systems. Her interests include database design, datawarehousing and data mining.

Brenton Williams
Information Technology Division

Brenton graduated from Adelaide University with an honours degree in computer science in 1990. Since then he has been employed at DSTO gaining experience in the domains of text retrieval, database management and data visualisation.

Greg Chase
Information Technology Division

Greg Chase graduated with honours from Flinders University in 1977. He joined the Electronic Warfare Division of the then Defence Research Center Salisbury in 1978. He participated in the development of electronic battlefield simulations and electronic warfare command and control systems. He joined the Information Technology Division in 1989 where he has been involved in work on Hypermedia systems, Management of Geographic Information and more recently Information Extraction.

Contents

1. Introduction

Defence agencies have a requirement to bring together information of different classifications from differing sources. This is particularly true of the databases used in these agencies. Security considerations are paramount to addressing data transfers between differently classified databases. Currently there are no accredited commercial products that comprehensively address this need[1]. The problem is further aggravated by the fact that these agencies have a mix of well entrenched legacy systems and more current commercial database systems that do not easily interact notwithstanding any security considerations.

The Starlight Interactive Link (SIL) [6] addresses the problem of connecting computers of differing security classifications in a trusted manner to enable a one way flow of data from the lower to the higher classified computer.

This report describes work undertaken to adapt the Starlight product to the database arena to address the problem of data sharing across databases classified at different security levels. The security levels can range from unclassified to top secret. Security considerations mandate that the flow of data be confined to being from the lower classified to the higher classified end. The Starlight product suite [6] meets this requirement.

Two options were investigated and prototyped in a demonstration system - the *ad hoc* and the contract based querying facilities. The former is useful for *ad hoc* querying of a lower classified database from a higher classified node. The latter addresses partial or full replication of the lower classified database at the higher classified node. The two methods are complementary and address different types of data sharing needs.

The next Section provides an overview of the problem and requirements. Section 3 discusses the two mechanisms identified to address the data sharing needs. Section 4 describes a prototype demonstration system in which the solutions were implemented. The report concludes with a brief look at some enhancements that have been identified should additional functionality become available in a future Starlight product line.

2. Background

Currently, data in Defence agencies typically resides in system high, standalone databases that span a range of security classifications from unclassified to top secret. Commercial multilevel secure database (MLSDB) products currently available are

[1] Existing commercial multilevel secure (MLS) database products can support differently classified data, within a restrictive range of security classifications. They typically offer lower functionality than their non-MLS versions [7].

not accredited to manage data sharing across the required range of security classifications [7].
Incompatibility between database systems used within and across agencies is not uncommon. The problem is further aggravated by the fact that some of the databases originate from external sources. The receiving agency therefore has little or no control in the choice of the database product or any other aspect of the system that it is given.

In view of these limitations, even if suitable commercial MLSDBs could be identified to enable data sharing, costs associated with switching from existing, legacy systems to a functionally richer commercial product may prove prohibitive. Additionally, the use of commercial MLSDB products could pose limitations on the autonomy of databases residing at the different sites.

The problem of data sharing across a differently classified environment has also been addressed by the US Naval Research Laboratory, using a Data Pump [1, 2, 11]. Their product uses different technology to Starlight and includes a low data rate back channel. There are some security concerns related to the back channel.

Our focus has been to use the Starlight product suite to address the problem of data transfer between different autonomous commercial database products residing in separate system high networks classified at different security levels. The solutions we identified and subsequently went on to implement rely on close cooperation between the donor and receiver sites. The data transfer mechanisms we explored are based on Starlight technology that was current at 1997.

3. Application of Starlight Interactive Link

3.1 System Architecture

Our solutions are based on the system architecture illustrated in figure 1. It comprises a lower classified system high network (L); a higher classified system high network (H); and, the SIL. The L and H systems are isolated from one another and can be running different commercial database products. The SIL establishes the link between the L and H systems in a trusted way, so the data flow from L to H does not compromise security. We address the problem of transferring data from a database on L to H.

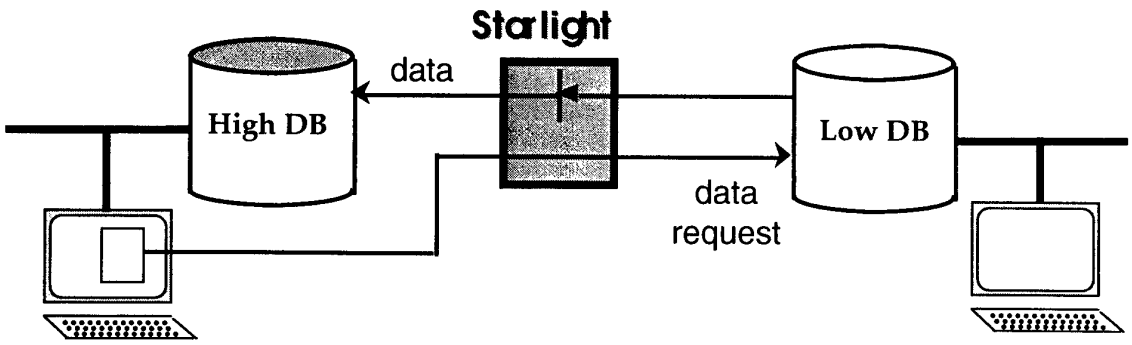

Figure 1: Interrogating a "lower" classified database from a "higher" classified system.

The solutions we identified for the data transfer problem fell into two categories:

- *Ad hoc* querying.
- Contract based data transfer.

The *ad hoc* approach aims to satisfy on-the-fly user requests for data from the L database. The contract based approach [3] is based on data transfer reliant on pre-specified contracts set up between the two sites.

Both approaches have an application depending on the data requirement of high end users. The *ad hoc* query is useful when a high end user needs to query parts of the L database at short notice, but involves the user having to manually deal with the data returned from the query. The contract based solution requires more planning beforehand, but allows the querying and data integration process to be automated.

3.2 *Ad hoc* Querying

An *ad hoc* query is an on-the-fly database query that a user on the H system executes on the L database. Two options are considered for executing *ad hoc* queries. The first involves a straightforward use of the SIL. The second is a more sophisticated approach.

3.2.1 Using SIL for *Ad hoc* Querying

This typifies the normal use of SIL. As supplied, the SIL can be used for querying a L database by a user on H who uses an x windows client running on a computer on L to access the L database (see figure 2). The results of the query will be displayed on the x windows client. The user must manually capture this information using the *cut and paste* utility and prepare it for integration into the H database.

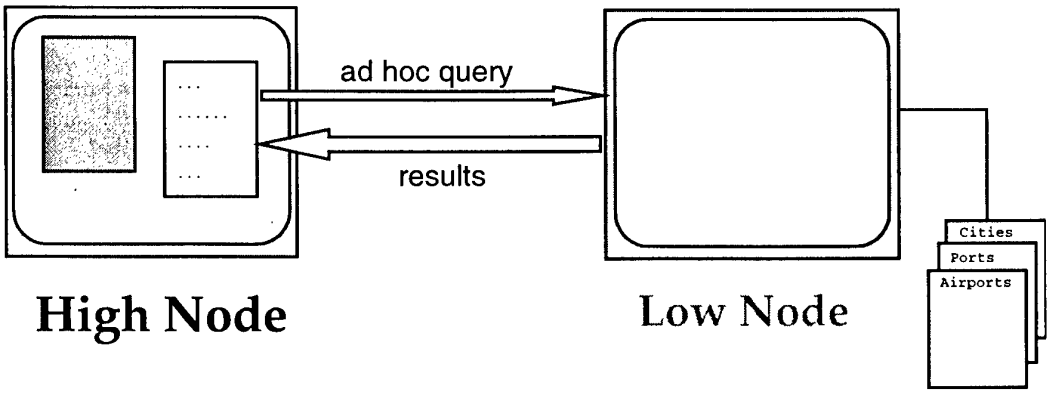

Figure 2: Simple ad hoc querying of an L database from the H node.

While this technique will allow a user to handle small amounts of data returned from the L database, it will be unsuitable for dealing with large sized datasets.

3.2.2 Extended *Ad hoc* Querying

A more sophisticated approach for *ad hoc* querying with the Starlight Interactive Link introduces a degree of automation to the querying process (see Figure 3). The application system for this approach comprises the following components:

On the L site:

- An *ad hoc query processor* agent.
- A *send file* agent.

On the H site:

- An *ad hoc file processor* agent.
- A *receive file* agent.

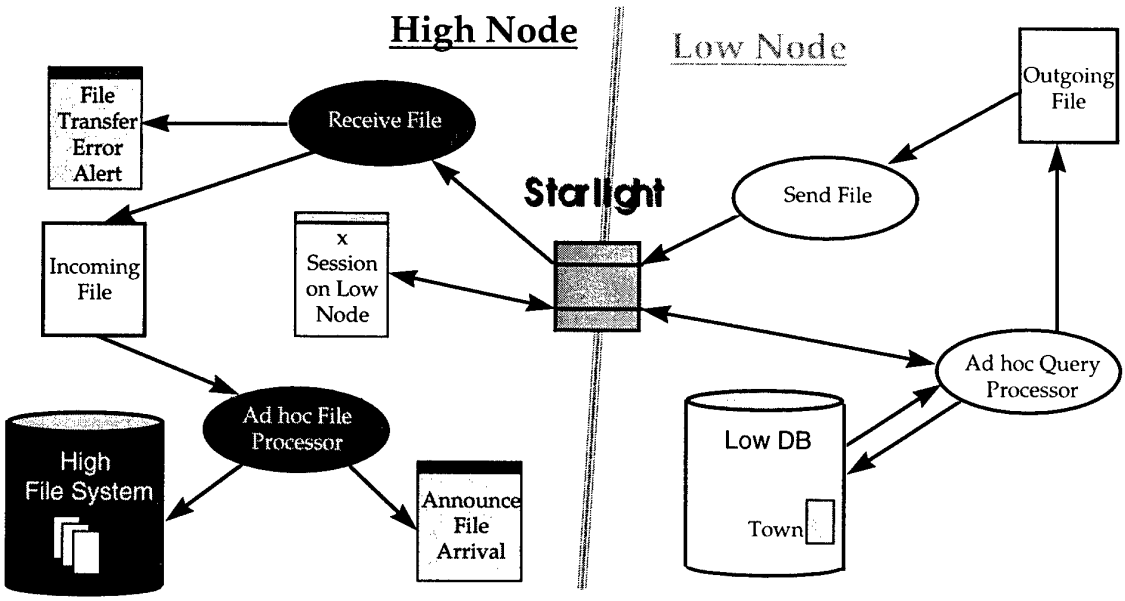

Figure 3: Extended ad hoc querying of an L database from the H node.

With the extended *ad hoc* approach, a user on the H node will start up an x windows client running on a computer on the L network and invoke the *ad hoc query processor* agent. The agent will prompt the user to enter the query to be executed. On entering the query, the agent will execute it and save results to a file. The file will then be sent across to the H network via the Starlight Interactive Link through the coordinated execution of the *send file* agent on the L system, and a *receive file* agent on the H system.

Upon arrival at the H site, the *receive file* agent will place the file in a nominated directory under its original filename. An *ad hoc file processor* agent will also run on the H system, interrogating the nominated directory on a regular basis to check for incoming *ad hoc* query results files. If any files are found, the agent will open an alert window and display the filename, query and resulting data from each file in turn. The user can browse through the query or data in the file and either save the file with a more meaningful filename to deal with later, or delete it as appropriate.

3.2.3 Uses of *Ad hoc* Querying

Ad hoc querying is useful for unplanned or on-the-fly querying of a low end database. It will require the L site to provide database access to a limited number of trusted users on the H site. The users will need training on the use of the L database system, as it may differ from their native database. They will need to be familiar with the schema of the L database or at least those parts of it that they are likely to want to interrogate.

When authorised high end system users have obtained the required data from the low end database, they may require assistance from their database administrator to

integrate it into the high end database. As the frequency and timing of *ad hoc* queries is unpredictable, the process of integrating query results with the high database is likely to place demands on the database administrator's time.

These costs are unavoidable if external data is to be accessed and integrated with the local database. Notwithstanding the costs, the *ad hoc* query method provides a useful way for trusted high end users to access external data from a low end database on demand.

3.2.4 Limitations of *Ad hoc* Querying

Ad hoc querying of a low end database by high end users is contingent on the users being granted authorisation to access the low end network and database. This involves them having an account on the L system and database. It is also reliant on high end users having a working knowledge of the interactive querying facility on the low end database. In addition, they need to be familiar with the schema at least of the subset of the database that they intend to query. This can place restrictions on the autonomy of the L database system as the L site would need to broadcast its database schema to the H site, and relies on the willingness of the two sites to cooperate closely.

Another limitation lies in integrating query results into the high end database. The *ad hoc* query option in both flavours requires the integration to be manually handled by users. They will need to transform the data from say, a report format into a form suitable for loading into the local H database. Additional costs could be imposed if they require assistance with the integration process.

3.3 Contract Based Querying

Contract based querying is based on the concept of using contracts to fully automate the transfer and integration of data from a lower classified to a higher classified database. The concept has been used in other work undertaken in the past [3], but lends itself well to this application. Using contracts to effect data transfers allows the transfer process to be fully automated. This includes automating the integration of the data into the local database upon arrival at the higher classified site.

3.3.1 Specifying Contracts

A contract is an agreement between the two sites to transfer some specified data from the lower classified database to the higher classified site and store it in the high database. Every transfer that can be effected, requires a contract to be put in place beforehand. Contracts are stored in the local database at each site. They can either be set up manually by the respective database administrators, or by using the Starlight Interactive Link. Every contract stored in the low end database will have a corresponding contract in the high end database.
A contract on the L site specifies:

- A contract identifier.
- A query to be executed on the L database. When the local database on the L site is an RDBMS for example, the query is expressed as a Structured Query Language (SQL) statement.
- A date/time when the query is to be executed. The query can either be a once-off query or be executed on a regular, periodic basis.
- Other ancillary information used by the agent that manages execution of the query on the L site.

A matching contract on the H site specifies:

- A contract identifier, that matches the identifier for the corresponding contract on the L site.
- The mechanism for dealing with the results returned from the L site. When the local H database is an RDBMS for example, an SQL statement specifies how the returned results are incorporated into the local database.
- Other ancillary information required by the agent on the H site to manage integration of query results returned from the L site.

3.3.2 Using Contracts to Transfer Data

Figure 4 illustrates the contract based approach to effect data transfers from the L to the H site. This approach relies on a number of agents and checks being in place at the two sites.

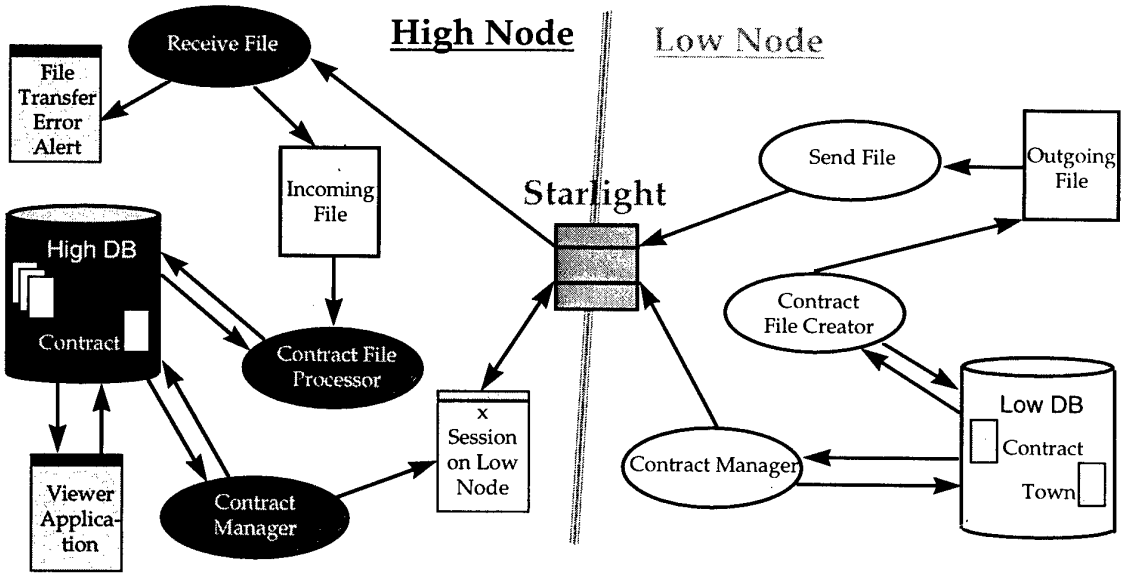

Figure 4: Contract based querying of the low end database from the high end site.

The main components will include:

On the L site:
- A *contract manager* utility.
- A *contract file creator* agent.
- A *send file* agent.

On the H site:
- A *contract manager* utility.
- A *contract file processor* agent.
- A *receive file* agent.

The *contract manager* utility at both sites will be used by database administrators to enter and manage contracts at their respective sites. In its simplest form, where the L site uses an RDBMS, the *contract* manager utility can be the interactive SQL interface to the RDBMS database product. The *contract manager* utility will allow contracts to be entered, deleted or modified at both sites. Contract management will require a coordinated effort by the database administrators at both sites. Email or other mechanisms across the Starlight Interactive Link can be used to coordinate and manage contracts.

Once contracts have been put in place at both sites no further user interaction will be necessary in the execution of contracts and assimilation of transferred data. The contracts allow this to occur automatically. Those users on H who need to alter existing contracts or establish new ones must first have these ratified through appropriate channels before their database administrator can effect the required changes.

A *file sender* agent running on the L site will regularly poll the local database for new contracts. If any new contracts are found, these will be added to a queue and tagged to be executed at the specified date/time. At the appropriate date/time, the query associated with a contract will be extracted from the contract record in the database and executed. Results of the query will be saved to a file. Some additional header information will be inserted into the file header, in particular, the **contract id** and error checking information.

The file will then be sent across to the H site by the *send file* agent in coordination with the *receive file* agent on the H site. On arrival at the H site, the *receive file* agent will store the file in a nominated directory.

On the H site, a *contract file processor* agent will regularly poll the nominated directory to check for incoming contract files. If one or more files are found, each will be handled in turn. For each file in the queue, the *contract file processor* agent will extract the **contract id** from the file header to identify the appropriate contract in the local database; use it to retrieve the appropriate database action statement stored with that

contract; and, execute the statement on the data in the incoming file to update the local database.

3.3.3 Uses of Contract Based Querying

Contract based querying has the advantage that the querying and assimilation of query results are both automated, making it easier for users on the H site once the contracts are established.

This technique allows database autonomy to be maintained at each site. As a consequence, the L site does not need to reveal its database schema to the H site. Also, no accounts need to be created for H users, as the contracts are entered at each site by the database administrator. The database administrator at the L site has complete control over the data being released to the H site, with the authorisation for release coming from appropriate channels. As both sites are able to maintain autonomy, they are more likely to cooperate in the data transfer process.

By enabling each site to retain database autonomy, the contract based method allows data transfer to occur even when the database products at the two sites are different and unrelated. For example, one site may use an object oriented database management system and the other a relational database system; or, one site can use an RDBMS from one vendor and the other site uses an RDBMS from a different vendor. The contracts will specify the format of the query results and how the results will be processed at the receiving site, so there is no requirement for the two sites to have the same or even related database products.

The contract based method provides a useful mechanism for part or full replication of the L database at the H site. This is particularly useful as the current Starlight product does not allow hand shaking to occur, ruling out the use of vendor supplied database replication tools which typically rely on two way communication between the master and slave sites.

Database replication involves duplicating relevant subsets of a master database at a slave site. In our case, the master is the database at the L site and the slave is the database at the H site. Management of the replication could take one of two forms - updating the slave database as soon as the master was updated; and, updating the slave database on a periodical basis.

The contract based method lends itself well to either form of database replication as long as the database being replicated is of a manageable size, and the data requirements of the H site users do not overburden the communication channel. The only limit imposed on the extent and frequency of database replication is the bandwidth of the channel.

3.3.4 Limitations of Contract Based Querying

The contract based method is dependent on the establishment of contracts before any data transfer can occur. The process of establishing contracts involves overheads, it is therefore not appropriate for on-the-fly querying. On-demand queries would be more appropriately handled by the advanced *ad hoc* query mechanism discussed in Section 3.2..

When the contract based technique is used for full or part database replication, or for piggybacked updates, it is necessary to have a mechanism that ensures that files are sent across to the receiving site in the order in which they were created at the sending site. This will guarantee database consistency between the master and slave sites (see Section 3.4).

3.4 Error Detection and Resolution

The extended *ad hoc* query and contract based methods discussed above rely on the *send file* and *receive file* agents to handle file transfer. Both agents use the User Datagram Protocol (UDP) to effect file transfers. UDP is non connection-oriented without data checksum, duplicate detection, sequencing, positive acknowledgment and flow control [4, pp 205ff]. This introduces the possibility of at least four types of errors:

1. File corruption.
2. Transmission of files in incorrect sequence.
3. Failure of transmission due to a broken communication link between the two sites.
4. Denial of service attacks.

Both the extended and *ad hoc* query and contract based methods need mechanisms to detect and resolve these errors. They are discussed in Sections 3.4.1-3.4.4.

3.4.1 File Corruption

File corruption can cause obvious problems at the receiving site. Checksums can be used to improve robustness of the file transfer mechanism by ensuring that files are transmitted uncorrupted.

For example, the *send file* agent can compute a checksum on the file contents and store it in the file header before transmission. On the receiving end, before performing other checks, the *receive file* agent recomputes the checksum and compares it with that stored in the file header. If a discrepancy is detected, the database administrator is alerted and further processing ceases. The database administrator must arrange for a re-transmission of that file and any later files as appropriate with the sending site (see Section 3.4.2).

3.4.2 File Sequencing

Transmission of files in incorrect sequence order can easily go unnoticed but have serious implications for database integrity and consistency if resulting updates occur in the wrong order; or, in the case of part or full database replication, the transactions are applied in the wrong order. The contract based method is particularly susceptible to this type of error.

In the absence of a two-way communication channel between the sending and receiving sites, the *send file* and *receive file* agents (see Section 3.3.2) can be modified to handle detection and resolution of file sequencing errors. For example, sequence numbering of files provides a mechanism to verify that files arrive at the receiving site in the order in which they are despatched from the sending site. Additionally, a copy of all transmitted files need to be retained at the sending site for a suitable time period to enable recovery from file sequencing errors. In particular, to enable re-transmission of files with sequence numbers equal to and greater than the missing sequence number.

3.4.3 The Link Status

A mechanism is required to monitor the integrity of the communication channel between the two sites, as the receiving site has no way of acknowledging to the sending site that the files that are transmitted are in fact being received.

For example, a dummy contract can be established which sends a file at a suitably small periodic interval, say 1 minute. The receiving site checks for arrival of the dummy contract file at the agreed interval. If the file is not successfully transmitted, then an error is raised at the receiving site and the database administrator is alerted, so appropriate action can be taken to re-establish the link.

3.4.4 Denial of Service

The receiving site, the *receive file* agent can be subject to a continuous transmission of files by a rogue L system user, causing denial of service. This problem is difficult to resolve. One way of limiting the damage from a denial of service attack could be to place space limitations on the directory where the incoming files are placed.

4. The Demonstration System

4.1 Demonstration System Architecture

A demonstration system was implemented to explore some of the options described above. The demonstration system was built around two database systems residing on system high networks:

1. A Sybase database residing on an unclassified (U) network, and
2. An Oracle database residing on a classified (C) network.

The Starlight Interactive Link was used to establish a link between the two systems in a trusted manner. The system architecture is shown in Figure 5.

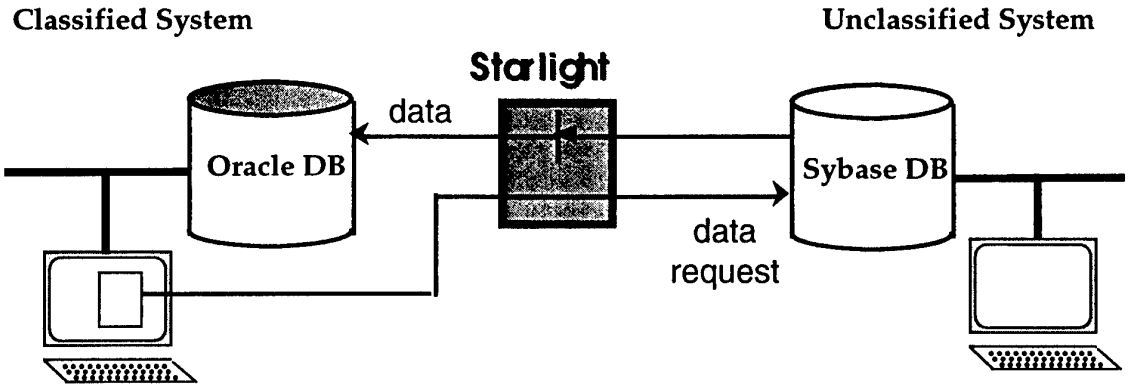

Figure 5: Interrogating an unclassified Sybase database from a classified system.

The demonstration system had the following components:

- A machine on an unclassified (U) network running a Sybase database.
- A standalone machine representing a classified (C) site, running an Oracle database.
- A Starlight Interactive Link was used to establish a one-way communication channel between the two systems.

The *ad hoc* querying and contract based data transfer methods were implemented as described in Sections 4.2-4.4. In both cases data flowed from the U to C site. It should be pointed out that the demonstration system was developed primarily to illustrate the concept of the two methods. While we were cognisant of error checking requirements for an operational system, these were not implemented in the demonstration system (see Section 3.4).

4.2 Implementation of *Ad hoc* Querying

Both the simple and extended *ad hoc* querying solutions were implemented in the demonstration system exactly as described in Sections 3.2.1 and 3.2.2 respectively. In the simple *ad hoc* querying system, standard RDBMS query tools were employed on the lower end workstation. Cut and Paste was used to integrate data at the higher end workstation.

4.3 Implementation of Contract Based Querying

A simplified version of the contract based method described in Section 3.3 was implemented in the demonstration system. In particular, contracts at both sites were managed by the database administrators, and some of the required error checking was omitted.
In the demonstration system, contracts were set up at both sites using the database specific interactive SQL tool, namely, **isql** for Sybase and **SQL*Plus** for Oracle. Alternatively, simple forms based applications could be developed to manage the contracts. Contract requirements at both the U and C sites are discussed in detail in Section 3.3.1.

A contract at the U site, stored in Sybase, contained:

- A contract identifier - a unique integer.
- A query to be executed on the L database - for example, a SQL statement such as:

```
select population, country from poptable where country in
('Australia').
```

- A date/time when the query is to be executed. This can either be specified as a regular interval, for example - `regular 3600` - for a regular hourly interval; or, it can be specified as a once-off specific date/time field, such as - `once HOUR:MIN:SEC dd/mm/yyyy`. The particular format for the date/time specification is configurable.
- Other ancillary information used by the agent on the L site that manages the query's execution, such as the contract date.

A matching contract on the C site, stored in Oracle, contained:

- A contract identifier, which matched that for the corresponding contract on the U site.
- The mechanism for dealing with the results returned from the U site. This was a SQL statement specifying how the results were to be incorporated in the local Oracle database such as a SQL *insert* or *update* statement.
- Other ancillary information required by the agent on the C site to manage the integration of results returned from the U site, such as the time the data was last sent.

The contract based solution implemented for the demonstration system is illustrated in figure 6.

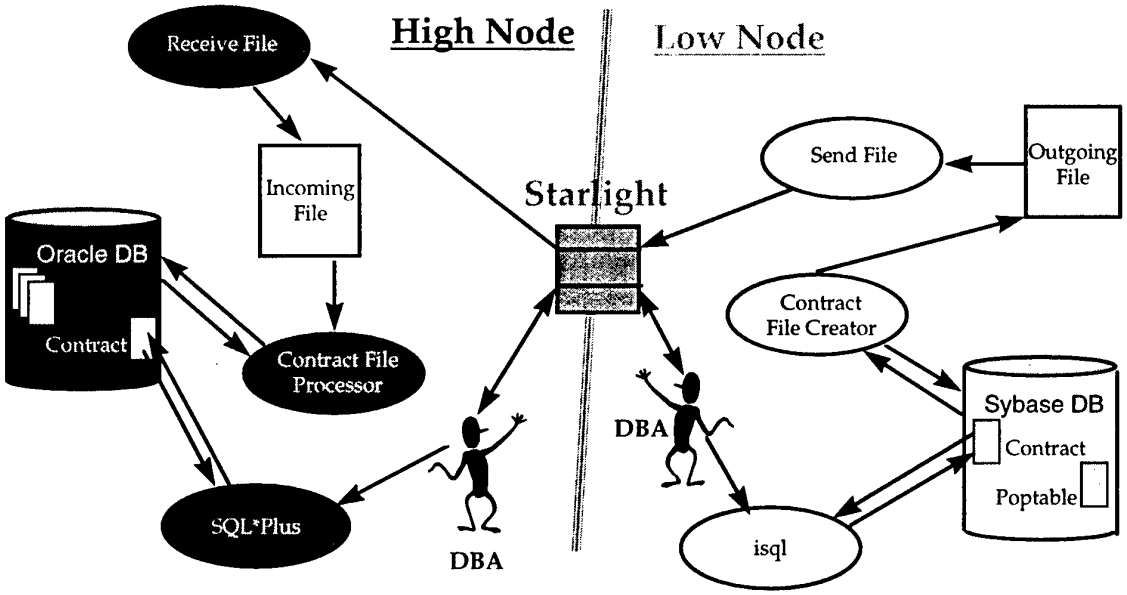

Figure 6: Using contracts to access an unclassified Sybase database from a classified machine.

The demonstration system was built to illustrate the concept of *ad hoc* and automated data transfer from an unclassified Sybase database to a classified Oracle database using the Starlight Interactive Link. While error detection and resolution (see Section 3.4) was not implemented for the demonstration system, it must form an integral part of any production system based on these methods[2].

5. Conclusion

In many areas of Defence, the requirement for accessing data stored in a lower classified database from a higher classified network is well acknowledged. Security considerations pose stringent constraints on the accreditation of software/hardware intended to support this type of data access. Currently there are no accredited commercial products that adequately address this need across the full range of security classifications required by Defence agencies.

The Starlight Interactive Link allows secure data transfer from a lower classified to a higher classified system in a trusted manner. We have shown how this can be extended to trusted database transfer.

The absence of a low data-rate back channel places severe limitations on the complete automation of the process. Significant gains can be made if the Starlight product suite

[2] We did not encounter any errors when testing the demonstration system, however it is necessary to detect and resolve errors that may arise out of unforseen circumstances.

was enhanced to include this facility. It is acknowledged that use of a back channel may not always be suitable for security reasons.

6. References

[1] M.H. Kang and I.S. Moskowitz. "A Data Pump for Communication", NRL Memo Report 55-95-7771 1995.

[2] M.H. Kang, Ira S. Moskowitz and D. Lee. "A Network Version of the Pump", Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1995.

[3] B. Williams, J. Das and P. Dart. "Development of a Distributed Heterogenous Database Testbed", Information Technology Division Divisional Paper, ITD-93-29, December 1993.

[4] W. R. Stevens. "Unix Network Programming", Prentice-Hall, New Jersey, 1990.

[5] J. N. Frosher, David M. Goldschlag, M. H. Kang, C. E. Landwehr, A. P Moore, I. S Moskowitz and C. N. Payne. "Improving Inter-Enclave Information Flow for a Secure Strike Planning Application", Proceedings of the 11'th Annual Computer Security Applications Conference, New orleans, Louisiana, December 1995.

[6] M. Anderson, C. North, J. Griffin, R. Milner, J. Yesberg and K. Yiu. "Starlight Interactive Link", Proceedings of the 11'th Annual Computer Security Applications Conference, New Orleans, Louisiana, December 1996.

[7] http://www.dsd.gov.au/epl - Defence Signals Directorate - Evaluated Product List - Section 7.

[8] Sybase SQL Server Reference Manual : Volumes 1 and 2, Server Release 10.0, 32401-01-1000-03, June 17, 1994.

[9] Oracle Corporation. "Oracle8.0.3 - Online Oracle8 Product Documentation", USA, 1997.

[10] Sybase SQL Utility Programs for Unix, Server Release 10.0, 30475-01-1000-04, February 1, 1994.

[11] M.H. Kang, I.S. Moskowitz and D.C. Lee. "A Network Pump", IEEE Transactions of Software Engineering, vol.22, no.5, May 1996, pp. 329-38, IEEE, USA.

Secure Database Data Transfer with Starlight

Jyotsna Das, Brenton Williams and Greg Chase

(DSTO-TR-0798)

DISTRIBUTION LIST

	Number of Copies
AUSTRALIA	
DEFENCE ORGANISATION	
Task sponsor:	
Director Operational Information System Development	1
S&T Program	
Chief Defence Scientist)	
FAS Science Policy)	1 shared copy
AS Science Corporate Management)	
Director General Science Policy Development	1
Counsellor, Defence Science, London	Doc Control Sheet
Counsellor, Defence Science, Washington	Doc Control Sheet
Scientific Adviser - Policy and Command	1
Navy Scientific Adviser	1 copy of Doc Control Sheet and 1 distribution list
Scientific Adviser - Army	Doc Control Sheet and 1 distribution list
Air Force Scientific Adviser	1
Director Trials	1
Aeronautical & Maritime Research Laboratory	
Director	1
Electronics and Surveillance Research Laboratory	
Director	1
Chief Information Technology Division	1
Research Leader Command & Control and Intelligence Systems	1
Research Leader Military Computing Systems	1
Research Leader Command, Control and Communications	1
Executive Officer, Information Technology Division	Doc Control Sheet
Head, Information Architectures Group	1
Head, Information Warfare Studies Group	Doc Control Sheet
Head, Software Systems Engineering Group	Doc Control Sheet
Head, Year 2000 Project	Doc Control Sheet
Head, Trusted Computer Systems Group	Doc Control Sheet
Head, Advanced Computer Capabilities Group	Doc Control Sheet
Head, Systems Simulation and Assessment Group	Doc Control Sheet
Head, C3I Operational Analysis Group	Doc Control Sheet

Head Information Management and Fusion Group	1
Head, Human Systems Integration Group	Doc Control Sheet
Head, C2 Australian Theatre	1
Head, Information Architectures Group	1
Head, Distributed Systems Group	Doc Control Sheet
Head C3I Systems Concepts Group	1
Head, Organisational Change Group	Doc Control Sheet
Task Manager	1
Author(s)	3
Publications and Publicity Officer, ITD	1

DSTO Library and Archives

Library Fishermens Bend	1
Library Maribyrnong	1
Library Salisbury	2
Australian Archives	1
Library, MOD, Pyrmont	Doc Control Sheet

Capability Development Division

Director General Maritime Development	Doc Control Sheet
Director General Land Development	Doc Control Sheet
Director General C3I Development	Doc Control Sheet
Director General Aerospace Development	Doc Control Sheet

Navy

SO (Science), Director of Naval Warfare, Maritime Headquarters Annex, Garden Island, NSW 2000.

Army

ABCA Office, G-1-34, Russell Offices, Canberra	4
SO (Science), DJFHQ(L), MILPO, Enoggera, Qld 4051	Doc Control Sheet
NAPOC QWG Engineer NBCD c/- DENGRS-A, HQ Engineer Centre Liverpool Military Area, NSW	Doc Control Sheet

Intelligence Program

DGSTA Defence Intelligence Organisation	1

Corporate Support Program (libraries)

OIC TRS Defence Regional Library, Canberra	1
Officer in Charge, Document Exchange Centre (DEC)	Doc Control Sheet & Distribution List
US Defence Technical Information Center,	2
UK Defence Research Information Centre,	2
Canada Defence Scientific Information Service,	1
NZ Defence Information Centre,	1
National Library of Australia,	1

Universities and Colleges

Australian Defence Force Academy	1
Library	1
Head of Aerospace and Mechanical Engineering	1

Deakin University, Serials Section (M list)), Deakin University Library, Geelong, 3217	1
Senior Librarian, Hargrave Library, Monash University	1
Librarian, Flinders University	1

Other Organisations

NASA (Canberra)	1
AGPS	1
State Library of South Australia	1
Parliamentary Library, South Australia	1

OUTSIDE AUSTRALIA

Abstracting and Information Organisations

Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
Materials Information, Cambridge Scientific Abstracts	1
Documents Librarian, The Center for Research Libraries, US	1

Information Exchange Agreement Partners

Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1

SPARES	5

Total number of copies: 61

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
DOCUMENT CONTROL DATA

1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)

2. TITLE

Secure Database Data Transfer with Starlight

3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)

Document	(U)
Title	(U)
Abstract	(U)

4. AUTHOR(S)

Jyotsna Das, Brenton Williams and Greg Chase

5. CORPORATE AUTHOR

Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury SA 5108 Australia

6a. DSTO NUMBER	6b. AR NUMBER	6c. TYPE OF REPORT	7. DOCUMENT DATE
DSTO-TR-0798	AR-010-880	Technical Report	March 1999

8. FILE NUMBER	9. TASK NUMBER	10. TASK SPONSOR	11. NO. OF PAGES	12. NO. OF REFERENCES
N8316/5/32	96/189	DOISD	24	11

13. DOWNGRADING/DELIMITING INSTRUCTIONS

N/A

14. RELEASE AUTHORITY

Chief, Information Technology Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for public release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS

Starlight
Data transfer
Secure Communications

19. ABSTRACT

There is a requirement in defence to connect secure computing facilities to unclassified information sources such as the Internet. This is particularly so in the database arena where data stored in differently classified databases must be shared. The Starlight Interactive Link, was developed to connect computers of differing security classifications in a trusted manner to enable a one-way flow of data from the lower to the higher classified system. Using the Starlight Interactive Link we have developed methodologies for secure data transfer between databases resident on computing systems classified at different levels, and constructed a prototype demonstration system using these techniques.