Nested Encryption Library for automated IPSec-based Anonymous Circuits Establishment

Hervé AIACHE, Matteo LAURIANO, Corinne SIEUX and Cédric TAVERNIER
THALES Communications S.A.
BP 82 – 160, boulevard de Valmy – 92 704 Colombes Cedex
FRANCE

Abstract: Nowadays, security and privacy are becoming two of the most critical issues for current and future generation of communications systems. Since the 80's, many efficient systems have been proposed to ensure flows anonymity, mainly derived from the so-called Chaum's Mix networks. However, these solutions suffer from a lack of integration with standardized IP approaches and therefore missed a wide adoption by the general public. This paper proposes an anonymous circuit establishment scheme derived from the powerful Mix networks concept and inheriting from the IPSec Framework. This solution has been implemented and experimented over a real testbed in view to analyze its impacts on multimedia flows end-to-end transmission.

Key-Words: Anonymous routing, Privacy, Traffic Flows Confidentiality, IPSec, Chaum's Mix.

1 Introduction

Nowadays, security and privacy are becoming two of the most critical issues for current and future generation of communications systems, as illustrated by the huge number of attacks identified and numbered day-by-day over the Internet. In fact, a lot of the services providers of the new economy are requesting (and in most cases, collecting) personal information in order to access to and to exploit attractive context aware and customized services (e.g. digital stores, location services or bank access). In this context, general users are required to give up to the providers, their personal information such as identity, bank account, location, preferences and so on. Therefore, attackers that only observe a network can acquire easily sensitive private information about users, which are not classically protected by information security techniques.

Moreover, even security problems of IT systems (i.e. integrity, authentication, confidentiality and non-repudiation) can be solved using encryptions, hash and MAC functions, these solutions are not really suited to solve privacy issues. In fact, private information protection requires others techniques that enable to masquerade traffic source and destination, traffic paths and the type of traffic the users generate. Basically, encryption techniques allow hiding "what users are sending" but not "who is sending", "who is receiving", "where the users are", "what the paths inside the network are" and "what the types of traffic are". This means that privacy protection is not only related to the protection of the content that the users are sending but also to their behaviours (and the associated traffic or routing information) on the network. In this way, privacy and security solutions requires today to protect enhanced personal information in view to not be disclose to unauthorized part, which could passively collect them for undesired, and illegal, purposes.

Since the 80's, many efficient systems (e.g. [4], [5], [3]) aiming at ensuring users' communications flows anonymity have been proposed to solve users' privacy requirements and are mainly derived from the so-called Chaum's Mix concept [4]. However, these solutions, mainly using proprietary anonymized content delivery services, suffer of a lack of integration with standardized IP approaches. This explains for part that they have not really been widely adopted by the general public.

Therefore, this paper presents an anonymous circuit establishment scheme derived from the powerful Mix concept and inheriting from the IPSec Framework. This paper is structured in three main sections: the section 2 details, for the first time, a functional view describing most of the proposed anonymous routing solutions (e.g. Tarzan[5], Mix[4], OR[6], TOR[3], MorphMix[8]), the section 3 details the Mix-like solution based on nested IPSec tunnels, the section 4 reports results obtained from real experiments analyzing the impacts of cryptographic algorithms on multimedia flows end-to-end transmission.

2 Functional view of anonymous routing approaches

Two main groups of privacy solutions exist : the peer to peer and the onion routing architecture which are mainly based on a node mix approach, but developed on distributed and dynamic environment for the first one (Tarzan, Crowds[7], MorphMix) and more centralized and fixed for the others one (Chaum's Mix-net,

mixMaster, Web-Mixes, TOR). The functional analysis of these various existing anonymous systems approaches shows that most of the fundamental anonymous routing solutions mainly rely on four main components and one building block defining specific transmission policies.

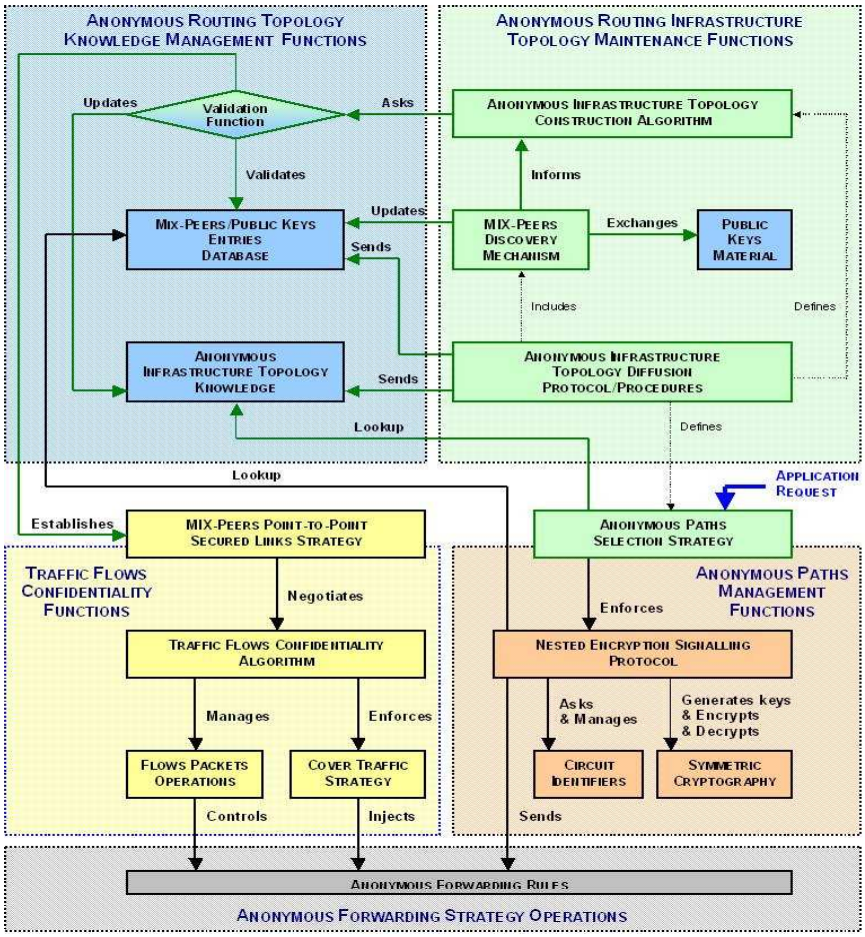

Fig. 1. Functional modelling of anonymous routing approaches.

Two components (sections 2.1 and 2.2) are more linked to the routing schemes (node discovery, topology establishment, trust validation) and the three others (sections 2.3 to 2.5) more linked to the forwarding plane. All these build blocks are illustrated in Figure 1 and are described in the following.

2.1 Anonymous Routing Infrastructure Topology Maintenance

This building block groups all the functions required to manage the MIX-peers access/leave inside the anonymous infrastructure, to exchange information on the MIX-peers in view to maintain the topology of the anonymous network. Note that, depending on the anonymous routing approach, these operations can operated statically based on procedures (or particular refresh based on a central server – e.g. TOR[3])or dynamically, such as in [2, 5], relying on specific peer-to-peer protocols.

The set of functions implementing the anonymous routing infrastructure topology maintenance defines in most cases a particular topology construction algorithm (e.g. CHORD, CAN or Gossip for P2P) in view to better control the infrastructure. At last, mainly associated to the anonymous infrastructure topology functions a particular anonymous paths selection strategy is implemented in view to choose the MIX-peers composing a given anonymous path, when applications requests it for anonymity service access.

2.2 Anonymous Routing Topology Knowledge Management

This building block integrates all the information structures (e.g. tables, databases) that are needed to know how the anonymous infrastructure is structured and what is the related cryptographic material (mainly public keys).

The table are initiated and filled during the MIX-peers discovery phase and updated by anonymous routing protocol/procedures. All these information will be used afterward to setup anonymous connections, as it will be described later.

2.3 Traffic Flows Confidentiality

This building block enables auto-configurable operations in order to masquerade traffic flows characteristics. It mainly relies on two specific sub-functionalities: modifications of the packet (i.e. padding and timing operations), and management of dummy packets.

It solves the problems of “correlation” and “timing” attacks. In most of the proposed approaches, these kind of problems would be solved by the definition of a particular Traffic Flows Confidentiality (TFC) algorithm.

2.4 Anonymous Paths Management

This building block commonly instantiates a pure or a variant of the well-known Chaum’s MIX [4] approach in order to protect the identities of the sender and/or the receiver (i.e. the MIX-peers identifiers or addresses). Mainly, this basic functionality aims at setting up an anonymous path based on the request issued by a specific MIX-peers set selection strategy (specified by the anonymous routing scheme). It usually sends a so-called onion-like signalling message (e.g. TOR[3], [6]) to pine the anonymous paths within the anonymous overlay infrastructure (through a specific format depending of the solution). Moreover, the

management of the anonymous paths is articulated around a particular management of circuit identifiers and is commonly and based on a particular nested encryption strategy (i.e. successive layers of encryption), which is in most cases elaborated through symmetric cryptography algorithms.

2.5 Anonymous Forwarding Rules

This set of rules groups a set of policies necessary at the transmission level to guarantee communication flows anonymity. This is required since the other functional components cannot masquerade all the information necessary to ensure properly the forwarding operation (e.g. MIX-peers identifiers or addresses change). Another usage of these forwarding rules can be also interesting to remove previously stored information on the communication flows and which are commonly recorded for optimization in current legacy communication stack (e.g. last IP headers).

This approach complements the IETF RFC 4303 that proposes the possible anonymization of the traffic by de-correlating the traffic useful for timing attack (Traffic Flow Confidentiality). An example of the solution illustrating the tunnelling technique is shown below in Figure 2.

3.1 Software design

The management of anonymous paths performed by the APM component can be decomposed into two distinct operations depending on the position of the Mix-node considered.

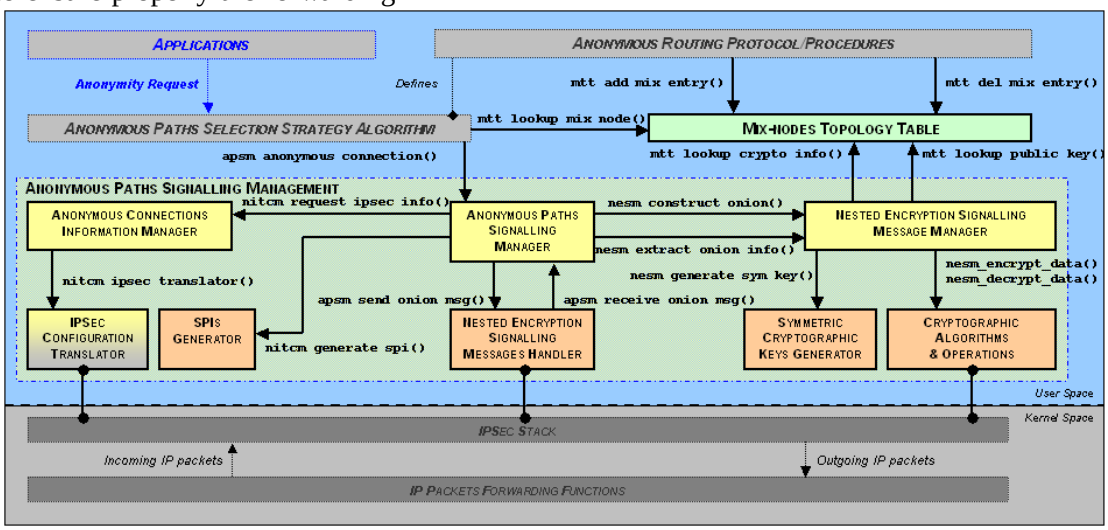

Figure 3. Software architecture of the APM module.

3 Design and Implementation

After fifteen years of efficient research and developments, the powerful obtained approaches have not really been integrated into Internet infrastructures. This is mainly due to the fact that solutions have not been built around standardized IP related security standards[1]. Therefore, to fill this gap, this paper specifies and designs a nested encryption circuit establishment natively inscribed in the IPSec framework and a software library to easily implement anonymous routing over IPSec. This solution answers to the passive attacks by traffic analysis and avoid source/destination linkability. The solutions refers to the function block Anonymous Path Management (APM) of the functional approach.

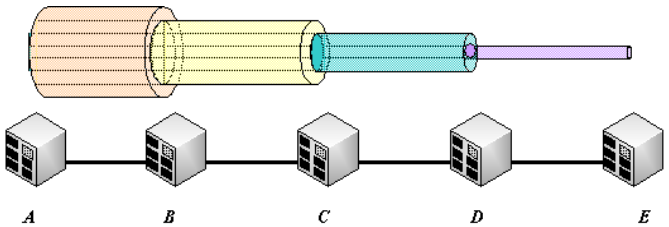

Figure 2. Nested Tunnelling Technique.

In fact, a Mix-node can initiate the setup (or the removal) of an anonymous circuit (i.e. by being the source node or acting for it) or a Mix-node can be an intermediary Mix-node of the selected anonymous path and, in this case, has to contribute to the establishment (or to the local removal) of the anonymous connection.

Therefore, this distinction is important to better understand how the APM software component (illustrated by the Figure 3) will act to implement these two different operations. In the case of an anonymous path setup, the Mix-node will have to generate all the necessary cryptographic material to establish the path, when in the case of an onion-like signalling message reception, it will have to enforce the corresponding IPSec tunnel configuration and to forward the peeled onion to the selected next Mix-node. These two different phases are explained in the following sections. The use of standardized and widely implemented solutions is recommended because they allow a simpler implementation and integration with the existing network infrastructures. Furthermore, the use of existing solutions is the best approach because their weakness and flaws are tested and well known so the problem is reduced to find a solution in order to solve these

systems' vulnerabilities and to adapt them to allow the untraceability and unobservability of traffic flows.

3.2 Operations overview

3.2.1 Setup operation of an anonymous path

After receiving a request for anonymity issued by an application, the *Anonymous Paths Selection Strategy Algorithm* asks for the enforcement of the elected anonymous circuit formed by several Mix-nodes, by calling the functions `apsm_anonymous_connection()` of the *Anonymous Paths Signalling Manager*. The IP addresses of the selected Mix-nodes composing the anonymous circuit are provided as input parameters of this function. Note that the order of the IP addresses is important since the anonymous connection will be established by contacting one by one each of the corresponding Mix-nodes in the same order.

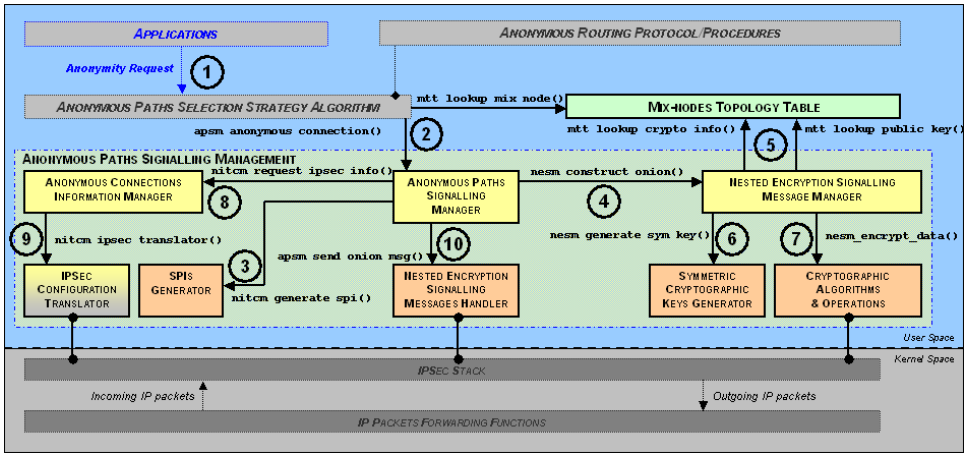

Figure 4. APM software components interactions to setup an anonymous path.

Then, the *Anonymous Paths Signalling Manager* demands the generation of several SPIs depending on the number of Mix-nodes involved in the anonymous connection, by calling the function `nitcm_generate_spi()` of the *Security Parameter Indexes Generator*. After this operation, the *Anonymous Paths Signalling Manager* requires the generation the appropriate onion-like signalling message: this is done by calling the function `nesm_construct_onion()` of the *Nested-Encryption Signalling Message Manager*. The list of the Mix-nodes IP addresses and the previously generated set of SPIs are given as input parameters of this function.

In view to generate the onion-like signalling message, the *Nested Encryption Signalling Message Manager* retrieves first the public keys (and the associated cryptographic algorithms to be used), corresponding to the list of Mix-nodes, by calling the function `mtt_lookup_public_key()` (and the function `mtt_lookup_crypto_info()`) of the *Mix-nodes Topology Table*. Then, the *Nested Encryption Signalling Message Manager* demands the generation of a set of symmetric cryptographic keys (that will be distributed to the set of Mix-nodes), by calling the function `nesm_generate_sym_key()` of the *Symmetric Cryptographic Key Generator*. At this stage, all the necessary cryptographic materials and information are known: the *Nested Encryption Signalling Message Manager* is now able to construct the onion-like signalling message. This operation is performed by calling appropriately the function `nesm_encrypt_data()` offered by the *Cryptographic Algorithms and Operations* block. Then, the onion-like signalling message and its homologue data (i.e. the same information but not encrypted) are then returned to the *Anonymous Path Signalling Manager*, as the result of the previous call to the function `nesm_construct_onion()`.

The *Anonymous Path Signalling Manager* informs then the *Anonymous Connections Information Manager* of this new anonymous circuit demand by calling the function `nitcm_request_ipsec_info()`. Note that the onion-like information data (i.e. the not encrypted one) is given as input parameter of this function. The *Anonymous Connections Information Manager* creates a new soft-state for this anonymous circuit demand, which is stored internally. Then the *Anonymous Connections Information Manager* enforces the corresponding nested IPSec tunnels configuration by calling the function `nitcm_ipsec_translator()` of the *IPSec Configuration Translator*.

Once the corresponding IPSec configuration has been enforced, the *Anonymous Paths Signalling Manager* sends the onion-like signalling message (the encrypted one) to the first Mix-node, by calling the function `apsm_send_onion_msg()` of the *Nested Encryption Signalling Message Handler*.

The Figure 4 illustrates how and in which order the APM software components interacts among each other to send the onion-like signalling message in view to setup an anonymous path.

3.2.2 Treatment of an anonymous path signalling message

When an anonymous paths signalling message is received by the APM daemon, it arrives inside the *Nested Encryption Signalling Message Handler* and is passed the *Anonymous Paths Signalling Manager* thanks to the handler called `apsm_receive_onion_msg()`.

Once received, the *Anonymous Paths Signalling Manager* peels a first layer of encryption of the onion-like signalling message by calling the function `nesm_extract_onion_info()` of the *Nested Encryption Signalling Message Manager*.

Then the *Nested Encryption Signalling Message Manager* extracts the required cryptographic materials and information thanks to the function of the `nesm_decrypt_data()` offered by the *Cryptographic Algorithms and Operations* block, using the private key of the Mix-node.

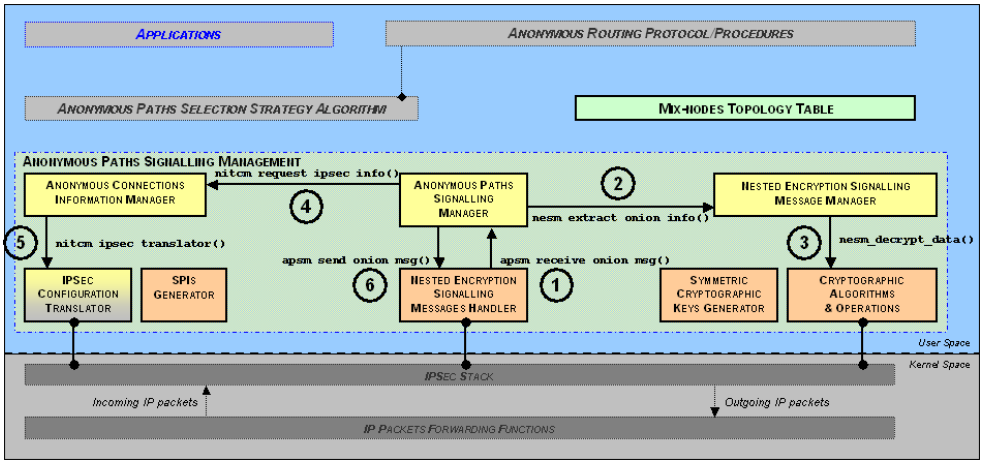

Figure 5. Treatment of an anonymous path signalling message by the APM software components.

After the performed decryption operations, the *Nested Encryption Signalling Message Manager* is able to separate the information related to the requested IPSec tunnel configuration (the symmetric cryptographic key to be used, the SPIs and the next Mix-node in the anonymous path) from the following of the onion-like signalling message that is made up of the other nodes' information (and encrypted with their public keys – so it is unable to decrypt).

The extracted IPSec tunnel configuration and the IP address of the next Mix-node are returned to the *Anonymous Paths Signalling Manager*, as the result of the previous call to the function `nesm_extract_onion_info()`. Note also that after the call to this function the onion-like signalling message, given as input parameter, has been peeled of its first layer of encryption and, in this way is ready to be sent to the next Mix-node. At this stage, the *Anonymous Paths Signalling Manager* demands then to the *Anonymous Connections Information Manager* to enforce this new anonymous circuit establishment request by calling the function `nitcm_request_ipsec_info()`. As previously, the *Anonymous Connections Information Manager* creates and stores a new soft-state for this anonymous connection. And asks the *IPSec Configuration Manager* to enforce the corresponding IPSec tunnels configuration by calling the function `nitcm_ipsec_translator()`.

Once informed of the IPSec configuration enforcement, the *Anonymous Paths Signalling Manager* sends the peeled onion-like signalling message to the next Mix-node, by calling the function `apsm_send_onion_msg()` of the *Nested Encryption Signalling Message Handler*.

The Figure 5 illustrates all these steps entering in the treatment of an anonymous paths signalling message.

4 Experimentations

This section describes the environment in which measurements have been done. The APM platform has been exploited in order to collect data. The platform is composed by five computers, as shown in Figure 6.

The five computers run Linux 2.6 and are connected in a chain. In this way four switched networks are defined. Three of them supports 100 Mbps, the fourth is 10 Mbps. The aim is to demonstrate if it possible to have secure communications for near real-time traffic and real-time traffic.

Five configurations were used: without tunnels, with one, two, three and four tunnels. Two different encryption algorithm were used: 3DES CBC and Rijndael CBC with its three different key sizes, i.e. 128, 192 and 256 bits. For each configuration measures have been done for packet size of 64, 128, 256, 512, 1024, 2048 and 4096 bytes.

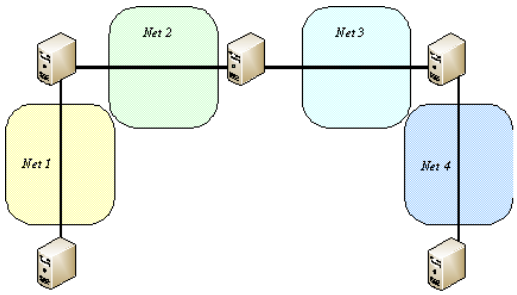

Figure 6. APM experimental platform.

In this section the measurements results are presented. In order to simplify the presentation two subsections are defined. The first one is about ICMP measures; the second one is about UDP.

4.1 ICMP measurements

The Round Trip Time (RTT) depending on packets size for a fixed encryption algorithm is presented. In each figure there are five curves; four of them represent the four configurations previously presented, the last one represents the case in which any tunnel is set up (the reference case).

Generally, the 3DES algorithm presents bad performances as shown in Figure 7; if packet size is not too large, i.e. minor or equal to 256 bytes, there are not great differences in the RTT between the configuration

with one and two tunnels. Relevant differences appears for packet size of 1024 bytes. In this case the four tunnels configuration presents a RTT of 50% higher than the case without tunnels.

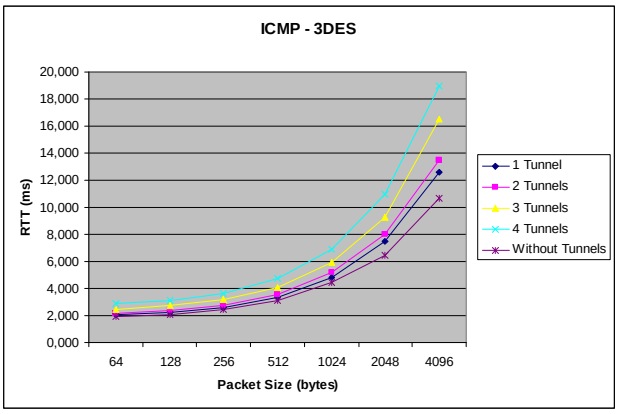

Figure 7. RTT vs. Packet Size using 3DES

Figure 8, Figure 9 and Figure 10 present the RTT using the Rijndael algorithm with, respectively, 128, 192 and 256 bits keys. It is important to underline that between these curves there are not big differences. Rijndael 128 is faster but differences are less than 0.3 ms.

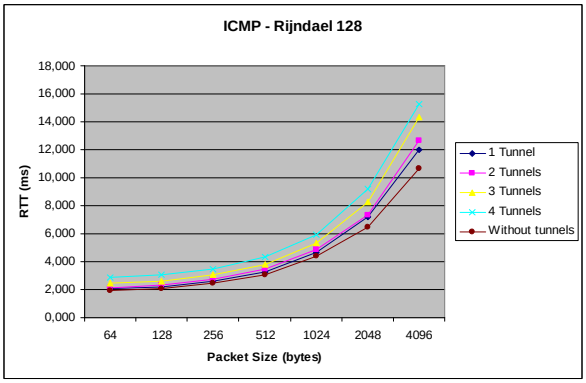

Figure 8. RTT vs. Packet Size using Rijndael 128.

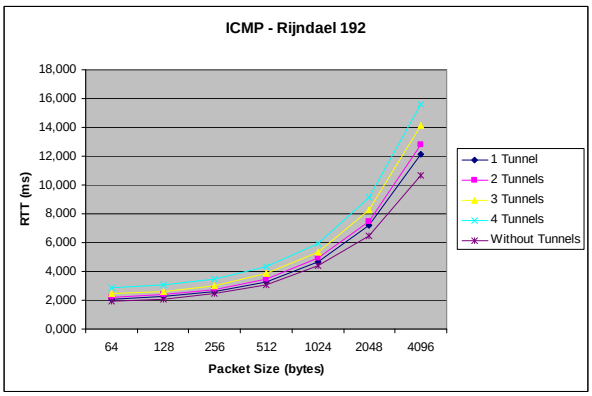

Figure 9. RTT vs. Packet Size using Rijndael 192.

This is an interesting result because it means that it is possible to increment the security of a transmission by the use of the longer key with a good level of performances. If packet size are not large, i.e. equal or greater than 1024 bytes, good values are reached.

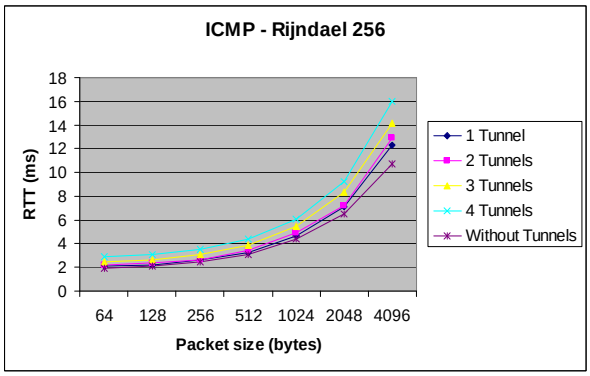

Figure 10: RTT vs. Packet Size using Rijndael 256

4.2 UDP measurements

This subsection discuss UDP jitter measures. These measures are evaluated for a fixed encryption algorithm and let's varying nested tunnels numbers. For each figure, different curves are available, the parameter is packet size. The knowledge of the jitter is interesting because high value or variation of this value can determine the difficulties to transmit real-time services.

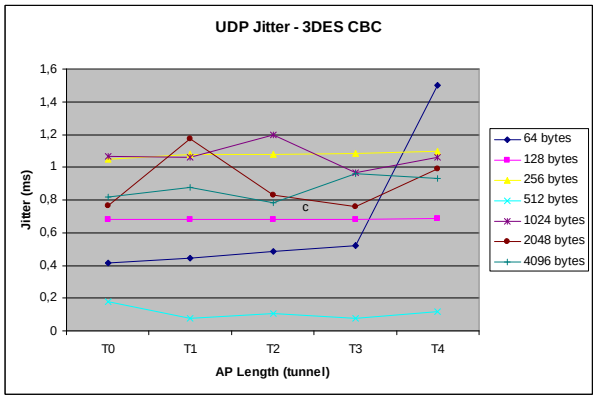

Figure 11. Jitter vs. Anonymous Path Length using 3DES.

Figure 11 shows jitter measured when 3DES algorithm is used. In this case 3DES is not the worst algorithm but a similar behaviour is observed with the other algorithm. Packets of 128, 256 and 512 bytes have the best behaviour because they are approximately constants when the number of tunnels changes.

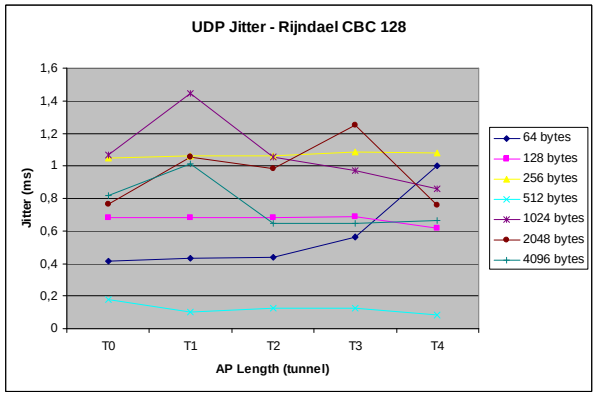

Figure 12: Jitter vs. Anonymous Path Length using Rijndael 128

The same conduct can be observed in Figure 12, Figure 13 and Figure 14. Packets of 512 bytes have the best performance; if 3DES or Rijndael 128 are used, they are the most regulars. In each case 64 bytes packets reach high value when three and four nested tunnels are established. The curves corresponding to 1024, to 2048 and to 4096 bytes packets are irregular. Generally, firstly they increase and then the decrease the jitter values when tunnels number increases.

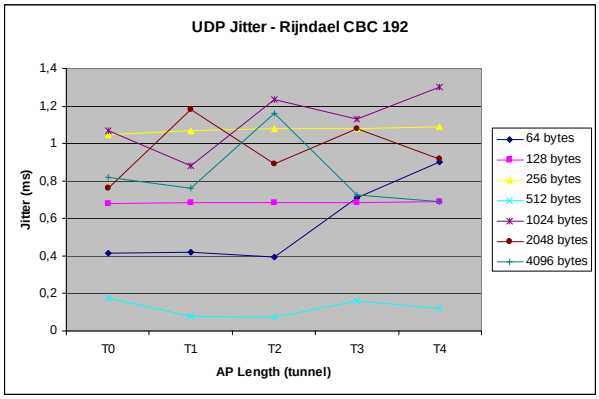

Figure 13: Jitter vs. Anonymous Path Length using Rijndael 192

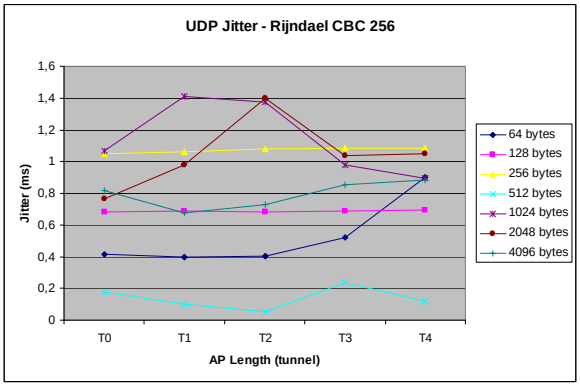

Figure 14: Jitter vs. Anonymous Path Length using Rijndael 256

The higher values are reached by packets of 1024 bytes when Rijndael is used and does not depend on the key length.

5 Conclusion and future works

This paper addressed and presented the critical main issues that IP networks have to face today: security and privacy protection. First, this paper extracted a functional modelling that applies to most of the anonymous routing solutions proposed in the literature.

Then, a Mix-like scheme based on a nested IPSec tunnelling technique has been proposed in view to ensure IP flows untraceability and unobservability. The particularity of the approach consists mainly in the management of anonymous circuits inside the standardized IPSec framework. Such a standardized environment constitutes one of the main constraints to the large-scale deployment and to the wide adoption of current anonymous routing systems.

Moreover, the proposed solution has been implemented and tested over a real experimental platform in view to characterize its impacts on multimedia flows (i.e. RTT and jitter) in function of the cryptosystem involved and depending of the anonymous circuit length.

Future works along these lines would include the performances and the security/privacy level improvements of the solution based on the implementation and tests of particular "message splitting" technique strategies. In this way, the IP packets sent by a given source would be transmitted to the same destination along different routes to make harder most of the passive attacks. In this case, the number of nested IPSec tunnels and the length of the anonymous circuits used for a given connection could be decreased if multiple route could be used and, in this way, the global performance would be increased, keeping in the same time a given privacy protection level. In other words, such a technique would improve the end-to-end performances due to the minor number of encryption/decryption operations calls. In addition to the efforts around the improvements of the solution, we plan to conduct studies on performance measurements with audio/video streaming applications over real testbeds.

6 Acknowledgement

We would like to acknowledge our partners from the IST FP6 DISCREET project with whom we have worked on the interactions between the Anonymous Paths Management and the Traffic Flows Confidentiality components. The DISCREET project was funded in part by the European Commissions' Information Society Technology 6th Framework Programme. It started in December 2005 and will end in February 2008. For more information visit the project web site http://www.ist-discreet.org/.

References:

[1] S. Kent, *"IP Encapsulating Security Payload (ESP)"*, IETF RFC 4303, December 2005.

[2] M. J. Freedman, R. Morris, "Tarzan: A Peer-to-Peer Anonymizing Network Layer", *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, 2002.

[3] R. Dingledine, N. Mathewson, P. Syverson, "TOR: The Second-Generation Onion Router", *Proceedings of 13th Usenix Security Simposyum* , August 2004.

[4] D. Chaum, *"Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms"*, Communications of the ACM, Vol. 24 Number 2, February 1981.

[5] M. J. Freedman, E. Sit, J. Cates, R. Morris, *"Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer"*, Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS02), 2002.

[6] D. M. Goldschlag, M. G. Reed, P. F. Syverson, *"Hiding Routing Information"*, Workshop on Information Hiding, Cambridge, UK, May 1996.

[7] M. K. Reiter, A. D. Rubin, *"Crowds: Anonymity for web transactions"*, ACM TISSEC, June 1998.

[8] M. Rennhard, B. Plattner, "Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection", 2002.