A Deployment Model of DNSSEC: Defining Problems and Solutions

Souleymane OUMTANAGA, Boubakar BARRY, Tiemoman KONE and Tanon Lambert KADJO

Institut National Polytechnique Houphouet Boigny, Laboratoire de Recherche en Informatique et Télécoms, BP 475Abidjan 08, Côte d'Ivoire

Université Cheikh Anta Diop, Faculté des Sciences et Techniques BP 5353 Dakar-Fann, Sénégal

Université de Cocody Institut de Recherche en Mathematiques Appliquées, BPV34 Abidjan 01, Côte d'Ivoire

Institut National Polytechnique Houphouet Boigny, Laboratoire de Recherche en Informatique et Telecoms, BP475 Abidjan 08, Côte d'Ivoire

Summary
The DNSsec protocol constitutes one of the solutions of DNS architecture's security. Unlike the other solutions, it represents an extension of the standard DNS. However its deployment is not without difficulty. In this work, we emphasize some problems involved in the progressive deployment of DNSsec, with the inherent risks in the use of public key cryptography. Proposals making it possible to solve these problems are provided.

Key words:
DNSsec, DNSsec deployment, PKI, Cryptography.

1. Introduction

The exponential development of the activities taken into account by the current Internet and with them, the plethoric number of Net surfers with various intentions, endangers all the Internet system. This explosion is mainly due to the progress of the Web. This latter is completely based on the mastery of the naming system: DNS. However like all the primitive protocols of Internet, it is not equipped with means allowing to face this evolution and also to defend itself against the multiple attacks of the Net surfers. DNS constitutes sometimes the ideal target for asphyxiation of Internet [1] [2] [3].

To mitigate this lack several works, were undertaken for its consolidation. The most known are DNSsec (*Domain Name System Security Extensions*) [4] [5] and the TSIG [6] [7].

If these various proposals are theoretically satisfactory, their deployment constitutes a challenge. The particular case of DNSsec constitutes the framework of this article.
Thus, we will present in section II, the extension of DNS security called DNSsec. It will be followed by the some problems description linked to the progressive deployment of DNSsec, its implementation compared to DNS, on the risks related to the use of public key cryptography in section III. Section IV is devoted to the prospects for the resolution of problems previously mentioned.

2. Security of the Naming System

In its original design, DNS did not take into account any security system. That situation leaves this protocol vulnerable. To solve these problems, DNSsec was developed [4]. It uses cryptography to protect DNS traffics.

2.1 The Public Keys Cryptography

The DNS constitutes a public utility. What does not make it possible to include all the cryptographic functions within the protocols ensuring the security of the system. Thus, the function of confidentiality of data is not recommended. The used functions are:

- integrity of the data
- authentication of the data and the data sender
- non repudiation

2.1.1 The asymmetrical encryption algorithm

The principle of the asymmetrical encryption algorithms is based on a pair of keys (key public and private key) [8] [9]. A message encrypted by public key (known of all its correspondents) is exclusively deciphered by using its corresponding private key (kept secret).

2.1.2 The hash function

To ensure the authentication of entities in communication or data integrity, the encryption algorithms and other tools are combined. One of these tools is the condensing or the hashing. Hashing (or condensing) is a function which transforms a variable size entry into a fixed size exit called the hash value. This hash value is the imprint of the initial message.

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

By taking a message M and its hash value H, it must be very difficult to be able to find a message M' such as the hash value H' of M' is identical to H.

2.1.3 The digital signature

The role of a digital signature is at the same time to detect the sudden modifications by the data in the course of transmission, but also to identify who sent them. It can moreover ensure the non repudiation of the transmitted messages.

The signature algorithms allow the creation and the checking of the signatures. For this, asymmetrical coding techniques are used [8] [9].

2.1.4 The certificate

When two entities have certificates, they make use of them to exchange their public key. It is a third of confidence in which the various involved parts have confidence which issue the certificates. The certificate contains, at least, an identifier and a public key of the entity to which it belongs, as well as the signature of the Certification Authority (CA) [8] [9].

2.2 The DNS Security Extensions (DNSsec)

The structure of DNSsec [10][11] is the same one as that of DNS. However DNSsec adds some improvements. Each zone has a pair of key (public key and private key). The public key of the child zone is signed by the private key of the parent zone, except the root which is signed by itself. A zone can also require a CA (Certification Authority) to sign its key. If we rely on a parent zone (or CA) we will also rely on the child zone signed by this relative. This forms the chain of trust. If DNSsec is globally deployed on the Internet, it will provide a universal mechanism of distribution of keys for all the entities.

Note: The RRset term which will be used in the rest of the document indicates whole resource records having the same name, the same type and the same class.

2.2.1 *The news Resource Records (RRS)*

DNSsec respects the backward-compatibility with protocol DNS: all the new objects required by DNSsec follow the RRs format of DNS, and its messages remain identical. In DNSsec, a zone is considered protected if there is at least a signature for each resource record except the NS records and the “A” records.

DNSsec required the installation of new resource records (RRs). These RRs has the same structure of course that RRs traditional and will differ by the format from the information which they contain. These new RRs are described in details in RFC2535[11][12].

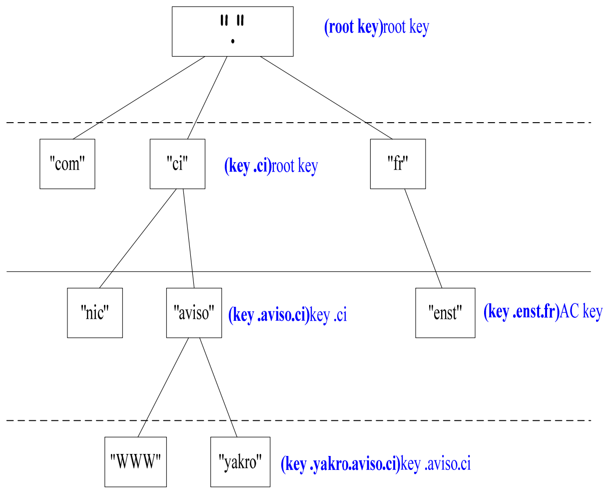


Fig. 1: DNSSEC Tree, **(X) y**, *key X is signed by the key y*

2.2.1.1 The Resource Record Key (Key RR becomes DNSKEY):

DNSKEY [11] [12][13] stores the public parts of the pairs of keys. They can thus be recovered by traditional resolution DNS each time we need to carry out checks of signature. It was considered judicious to distinguish the keys with which one will sign information of a zone, keys which will use to establish the chains of trust. We use the term ZSK (Zone Signing Key) to indicate the keys which will sign RRsets of a zone. The other type of key is called KSK (Key Signing Key). The KSK will be thus the intermediate links between the zones: the KSK of a zone is authenticated by the parent zone, and will be used in the child zone only to sign the KEY RRset (which contains the KSK and the ZSK(s) in Fig.6). That results in partitioning the levels of security, local and global by the use of distinct keys and in particular to facilitate the operations of rolling keys: if a zone wishes to change ZSK, it will not need to refer to its parent zone since the KSK will remain always authenticated. The picture below presents a DNSKEY and its flags fields.

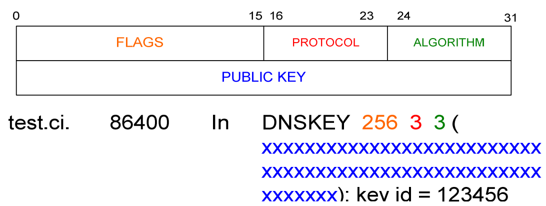


Fig. 2: Example of DNSKEY

2.2.1.2 T*he resource record SIG (SIG RR)*

A SIG RR stores the signature of a given RRset by a given key; each RRset of a zone will be accompanied by as many signatures as there are active ZSKs in the zone.

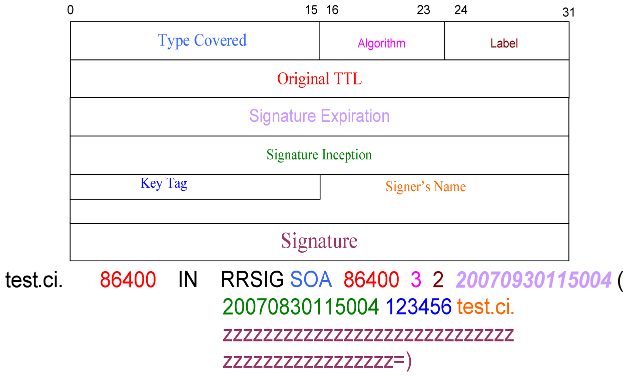


Fig. 3: Example of RRSIG

The DNSKEY RRset as far as is concerned is signed by ZSKs and KSK. All these signatures have a validity period of time outwards which they are considered to be invalid (picture 3 gives a RRSIG and its flags fields).

2.2.1.3 *The resource record NXT (NXT RR become NSEC)*

In protocol DNS, when a request related to a resource or a domain name which does not exist, the answer contains only one error code in its heading and this heading is not protected. The NSEC Record was created to be able to check the negative answers [11]. A NSEC record contains information necessary to the identification of the existing records for a given name, as well as the next existing name in the zone. Both information are enough to prove that a record or a domain does not exist. When a resolver receives a NSEC record, it checks its signature, and then looks at the associated name. If it is not the name which it asked, but that the required name is in an interval given name, the resolver deduces that the required name does not exist. The NSEC record which permits to check that a given domain does not exist is called the covering NSEC. When a resolver receives a NSEC record for the name which it asked, it checks in the Bitmap field if the type of required record is present. If this type is non-existent then the required record does not exist. The NSEC Record being protected by a digital signature, we can prove that the data which it contains were not modified. An example of NSEC record is shown in Fig. 4. This record specifies that there is no name between **test.ci** and the next name zone1.test.ci**,** and that type of existing records for the **test.ci** are NS, SOA, RRSIG, NSEC, and DNSKEY.

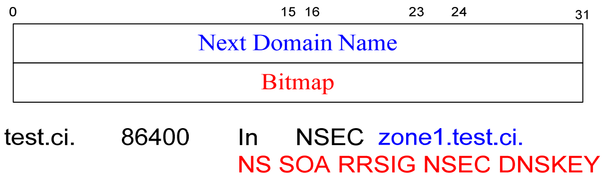


Fig.4: Example of recording NSEC

2.2.1.4 The resource record DS (DS RR)

The DS (Delegation Signer RFC4034 [11]) is a record concerning a child zone, but localised in the parent zone thus creating a secured link between both zones. It contains the hash value of the KSK of the child zone and is signed by the ZSK of the parent zone. Fig.5 shows a DS record and its flags fields.

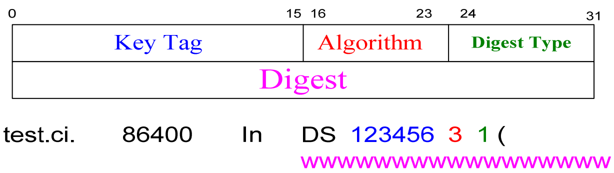


Fig.5: Example of recording DS

The key of the parent zone authenticates the DS of the child zone, which authenticates the KSK of the child zone, which itself signs the KEY RRset of the child zone. Thus the protected delegation is active. The model of a chain of trust on three levels (child zone, parent zone and grandparent zone having each one a ZSK and a KSK) is the following:

1. RRsets of the child zone are signed by the ZSK of the child zone;
2. The ZSK of the child zone is signed by the KSK of the child zone;
3. The KSK of the child zone is authenticated by the parent zone by generating corresponding RR DS and by including it in the file of parent zone;
4. This DS is signed in the parent zone by the ZSK of the parent zone;
5. The ZSK of the parent zone is signed by the KSK of the parent zone;
6. The KSK of the parent zone is authenticated by the DS corresponding in the grandparent zone;
7. This DS is signed by the ZSK of the grandparent zone.

Thus, if a resolver is configured with the KSK of the grandparent like trust key, it will be able to check information of the zone girl by building the chain of trust described previously. In Fig.6, there is an example of a chain of trust linking the trust key (KSK) of zone ci to the records (RRsets) of the grand-child zone **server.test.ci.**

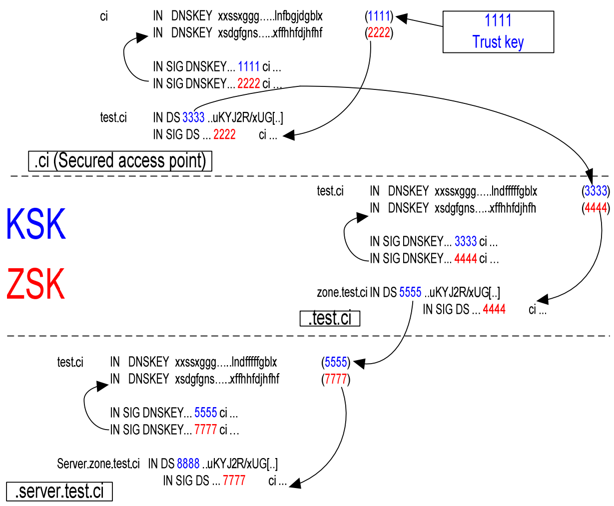


Fig.6: Example of a chain of trust

2.2.2 The Name Resolution with DNSsec

The principle of name resolution remains the same one for DNSsec. One of the constraints when creating the security extensions of DNS was the compatibility between DNS and DNSsec: equipment not including DNSsec must be able to carry out name resolution of DNS without having problems. The functioning of old equipment, faced with records which it does not know, is simply to ignore them. The scheme of exchange of the messages remains the same one, as well as the communicating entities [14].

The contents of the answers are more important in size because it contains additional records. A secured names server (having a signed zone file) will include in its answers the necessary cryptographic material, such as its DNSKEY records. Moreover, a record is always sent with its associated signatures.

In same way, if the answer is a delegation, the secured server will send in answer appropriate DS records and their signatures. If at given moment of the secured resolution, if some secured records are missed cache server or in the resolver, a specific request will be sent to recover the missing records.

3. Problems Linked to the Implementation and the Deployment of DNSsec

The goal of the deployment of DNSsec is to allow, with a trust key to access in a secure manner to all information of the tree. It would be necessary for all the nodes of the tree to be secured. However, in view of the number of zones and delegations to be secured, the deployment of DNSsec could be done only gradually [11] [14]. Thus, some secured parts of the tree will keep close to others non secured (Fig.7). We call secured islet an under-tree of the DNS tree in which all the zones and delegations of the under-tree are secured. The information contained in these islets could be considered sure by resolvers having the KSK of the top zone of the islet configured as trust key.
Fig. 7 shows three zone categories:

- None secured zones: they are not signed (**fr**).
- Local secured zones: they are signed but not connected to their parent zone by a secured delegation (demo.ci). In this case we can check the veracity of the information only if we rely on the KSK of that zone.
- Global secured zones: they are signed and we can access them by means of secured delegations (zone1.test.ci). There are several levels of global security according to the highest point towards which we can go up in the tree by means of secured delegations (to the root).

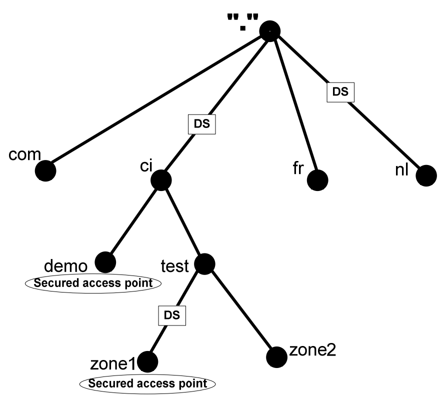


Fig.7: Example of partially secured DNS tree

In the example of Fig.7, we notice that RRset given will be considered sure; non sure or erroneous by a resolver according to its secured access point. Indeed, we notice that there is no DS for the zone "test.ci", a resolver configured with the key of the root as trust key will consider the zone "test.ci" like all the downstream zones like non-sure, and even if for example the zone "zone1.test.ci" is signed. On the contrary a resolver configured with the key of "test.ci" will consider the zone "zone1.test.ci" as sure since a DS authenticates it in "test.ci".

At this level of the DNSsec deployment, even if RRset is considered non sure we will be able to compare to its belonging zone decide to trust it what means that there will be some non secured zones which will have the possibility of making data of their zone circulate in the tree. At the same time, some resolvers will also reject certain data since they will consider them erroneous. With such conditions all the resolvers will not have access to all information which they need, since for security reasons they will make selection of information. In addition to that the not signed zones will be always victims of attacks related to the faults of the DNS: Then if the a DNS sever is spoofed, in a zone where RRsets are considered acceptable by some given resolvers, these resolvers will consider these false information acceptable, hence good to exploit. Whereas actually these data are false.

In fact, if the progressive DNSsec deployment does not take all these factors in account, the system of name resolution of will have some functioning problems as well as Internet.

We should not forget that except the role that the DNS plays in today, its success is also based on the simplicity of its implementation as well at the Server as at the Client. Indeed DNSsec is based on the use of public key, this is materialized by the addition of cryptographic material and some new records in the zone file. So there is an increase in the size zone file hence increase of answers which can involve the use of TCP more often than UDP (the DNS speed traffic decreases).

Besides the zone file becomes cumbersome for the administrator. If a zone has three pairs of keys and thus three DNSKEY records, after signature of the zone file, that will represent six records: three DNSKEY RRs (the DNSKEY RRset) and three RRSIG RRs, Fig.8 gives an example of it.

A part from the possibilities of decrease of the DNS speed traffic and cumbersome of the zone files, another factor is to be taken into account in the deployment of DNSsec. This factor is the complexity of the mechanisms of implementation (configuration, keys generation, keys actualization …) of DNSsec compared to DNS [4]. The procedures are very manual, which could rise some problems, even for initiates. What would lead to a slow deployment for certain zones or expensive for those which will call upon skilled in the domain.
We saw that the force of DNSsec resides on the use of crypto system with public key (private key/public key). In such a system, when a key is compromised, that can constitute a serious fault for tree architecture like that of DNSsec. Indeed, if a zone key is compromised, all the under domain of the zone is threatened and consequently all the tree through systems of update of the Cache of the name server. There are many ways of compromising a key. The first is the cryptanalysis. A hacker obtains the private part of a key thanks to mathematical knowledge and cryptographic material generated by the key or sometimes by using faults in the protocols generating the keys or the digital signature. We know, concerning DNSsec that it is advised to preserve the private keys in a sure place (disconnected from any network). The second way of compromising a key is, in case of non respect of this instruction, to have access of these keys by the network and of copying them by spoiling all the security measures. The third possibility is when attacker has a physical access to the private keys: it could concern an administrator (dissatisfied or having bad intentions) system or network of a given entity.

A compromised key allows the attacker to create false delegations and also to false answers acceptable or false records as if they were completely licit and correct. For more details on possible attacks with compromised keys in DNSsec [5].

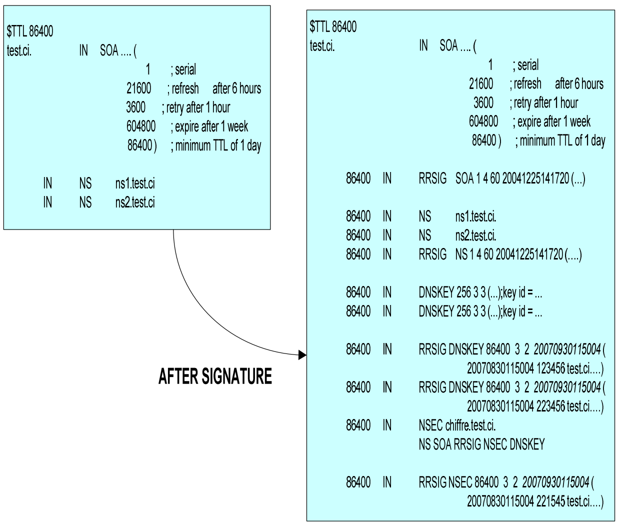


Fig.8: Signature of a zone file.

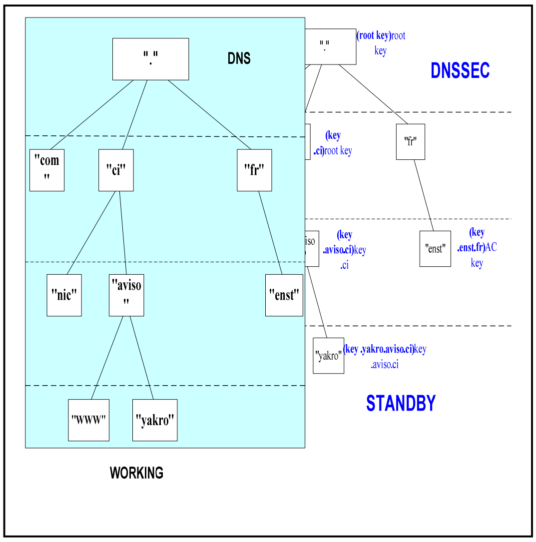
4. Prospects for a Made Deployment

Progressive deployment of DNSsec, as we saw above could create an ill-functioning of the name system resolution. We note that up to date all the solutions suggested to solve the problem of the cohabitation between secured islets and not secured one is based on whole deployment mechanisms of DNSsec. Since, if the DNSsec is partially deployed, its efficiency will decrease because of non secured zones which will be always subjected to the faults of DNS; unless finding automatic deployment mechanisms of DNSsec, who would make all the zones and the delegations of the tree signed in rapidly. However, up to now the solutions are slow and meanwhile our name servers are victims of all sort of attacks. Therefore groups and companies, which are victims of these attacks continue to lose important amount of money for some of them and technological innovations for others. This deployment should not be progressive, if we want DNSsec to be efficient. For this purpose, we could consider the possibility of creating a parallel architecture (GHOST) as the picture below shows it. This tree will be entirely secured (based on DNSsec) but in furtive mode, until a date fixed by the organizations planned for this project. Date from which any zone not having made the necessary arrangements to join the parallel architecture will not have access to the secured tree, since its RRs will be considered erroneous by the system. To carry out this system each zone will have to create a server dedicated to this project. This server will be a copy of this latter in work on the DNS architecture but connected to the DNSsec project architecture to be able to carry out tests to ensure itself of the good evolution of the project. This solution is based primarily on the DNSsec efficiency which is based on a total reassurance system of the tree. The Project named "GHOST" will have three phases which are:

- *Phase I*: Launching, during which all the zones and domains concerned by the name system resolution should take necessary arrangement to join the project. At this level the beginners will be able to profit freely from the experience of those who have already tried out DNSsec.

- *Phase II*: Validation, during which all the operators will make tests (Attacks on the protocol, simple navigation…) on network GHOST to ensure itself of the good walk of DNSSEC.
- *Phase III*: Activation**,** at this level we are near close of the activation date of GHOST network, DNS tree is deactivated and the tree (DNSsec tree) which was at the experimental stage replace of old tree (DNS tree). Even if at this level there is still no automatic mechanism for the rolling of the trust keys, we will have the advantage of being broken in the manual configurations which from *Phase II* have done it as a matter of routine for all actors of the domain name security system.

The size of DNS messages increases in DNSsec, because of the new records which added to secure the data. However, the addition of compression mechanisms must be considered in order to maintain DNS answers size less than 512 bytes, to be able to profit from the speed of UDP. A good level of security requires efforts as well on the level of the configuration as at the budgetary level. Indeed, the job of administrator will increase; that is to say the management of rolling mechanism of trust key and also the zone file enquiries. As regards the rolling of trust keys, automatic proposals are seen. We can mention, the Automated Updates of DNSsec Trust Anchors published by M. St Johns [16]. Moreover, the heads of firms and the decision makers should give financial means to train their personnel on the evolutions of technologies of information and telecommunications, because the DNSsec is still ignored by many data processing specialists.
Until the deployment of DNSsec, the administrators will profit from other tools to guide them in the implementation of DNSsec. They will be able to also profit from formations, and the expertise of their fellow-members, who have already started to try out DNSsec.



We should note that cryptography with public key is based on the use of pair of key (private/public). Today it is very difficult to break a key using cryptosystem; the problem is how to conserve his/her private key. For this purpose we have several solutions such as:

- To keep its key outwards from the network which implements DNSsec, on a support or a machine this must be disconnected if possible from any network.

- To use some methods used by certain banks secure their administrator password and some of their keys; safety deposit box or left-luggage lockers.
- To reserve a special treatment to the administrators system so that they are not tempted with the reason of an ill treatment or an unspecified frustration to attack the system. Since the most dangerous attacks are those which come from a person who knows perfectly the faults and the operation of its victim.

Our aim is to make our private key inaccessible by a malevolent person. It is also important to note that, if by any means your private key is discovered by another person, DNSsec architecture is not responsible for that situation. Each user of the system must take every step to keep his/her private key secret in order to guarantee the integrity of the data which forwards on the tree.

5. Conclusion

We have in this paper shown how the extensions of security of DNS (DNSSEC), brought integrity to the data and the authentication thank to the use of digital signature. After that, we have presented the deployment of DNSSEC and shown some problems linked to its progressive deployment, its implementation and its force based on the use of public key. We have then given some prospects in order to face these problems. However we will be able while waiting for the deployment of DNSSEC, to turn to associations of protocol inter alia: DNS/IPSEC [22] to increase the security level on the present resolution names system.

References

[1] Wellington Nominum. *Secure Domain Name System (DNS) Dynamic Update*, (November 2000), RFC 3007

[2] Pierre BETOUIN, *Failles intrinsèques du protocole DNS*, (October 20, 2003) http://securitech.homeunix.org/dnsa/Article_fr.txt

[3] Gilles WATCHES, Bernard COUSIN. *Les faiblesses du DNS*, Université Rennes 1, (2003) http://www.irisa.fr/prive/bcousin/Articles/SAR-2003.pdf

[4] Projet IDSA. *Sécurisation du DNS: les extensions DNSsec*, Rennes, (2003) http://www.idsa.prd.fr/atelier-idsa/atelier1.5.pdf

[5] Bertrand Léonard. *Etude et resolution des problèmes de delegation de DNSsec*, (2003). ftp://ftp.irisa.fr/local/idsa/doc/livrable/L2/1/pdf/l21.pdf

[6] SUPINFO. *Introduction to the safety of DNS*, http://www.supinfo-projects.com/fr/2004/dnssec/1/

[7] Nicolas Notari, Jean-Philippe Pick, Mohsen Souissi: *Federator project G6-dnssec*, (Nov. 21. 2002)

[8] TA Quoc An. *DNSSEC et la distribution sécurisée de clef*, Institute de la Francophonie pour l'Informatique (November, 2004). http://210.245.52.197/rapports/stages/promo07/stage-ta_quoc_an.pdf

[9] Herve SCHAUER. *Introduction to cryptography*, (2001)

[10] Giuseppe Ateniese, Stefan Mangard, *A New Approach to DNS Security (DNSSEC)*. (Nov. 2001)

[11] Arends, R., Austein, R., Larson, Mr., Massey, D., Rose, S.: *Resource Records for the DNS Security Extensions*, RFC 4034, (2005)

[12] D. Eastlake, *Domain Name System Security Extensions*, RFC 2535, March 1999

[13] DNSSEC and Related Drafts (IETF): http://www.dnssec.net/drafts

[14] AFNIC. *Le système de nommage*, http://www.afnic.fr/ext/dns/html/seq4891.html

[15] Bertrand Léonard, Olivier Courtay. *Study of the support of DNSsec side waiter*, AFNIC, (February 2004)

[16] M. StJohns, *Automated Updates of DNSSEC Trust Anchors*, draft-stjohns-dnssec-trustupdate-01.txt, Juillet 2004

Souleymane Oumtanaga received the PhD degree in Computer Science from University Paul Sabatier of Toulouse, France in 1995. During 1999-2000, he stayed in LAboratoire de Recherche en Informatique et Mathématiques Appliquées (LARIMA) at INPHB (Institut National Polytechnique Houphouet Boigny), Côte d'Ivoire. Since 2000 he has been Head of Centre de Formation des Technologies de l'Information et de la Communication (CFTIC) of INPHB. Since 1990 he has also been the Head of the Network Information Center (NIC) of Côte d'Ivoire. He has been Professor in computer science since 2007 and he currently manages the Laboratoire de Recherche en Informatique et Télécoms (LARIT) at INPHB. His research interests include IP mobility, IP Network security, IPv6, Wireless Network, Mobile Networks.

Boubakar BARRY received his MSc degree in Nuclear Physics in 1986 and his PhD in Nuclear Electronics in 1989, both from Technical University of Dresden, Germany. From 1993 to 2003, he performed several research visits at the Institute of Telematics of the University of Karlsruhe, the Technical University of Munich and the Technical University of Dresden, all in Germany. Since 1989, he was lecturing Electronics and Computer Networks at the Faculty of Science and Technology of Cheikh Anta Diop University in Dakar, Senegal, before joining the Association of African Universities in 2006 as Coordinator of its Research and Education Networking Unit.

Tiemoman KONE received the PhD degree in Computer Science from Universite Paris 12 Val de Marne in 1993. Since 1997, he is researcher at Université de Cocody-Abidjan in Cote d'Ivoire and also member of LARIT (Laboratoire de Recherche en Informatique et Télécoms) since 2006.

Lambert Tanon KADJO is currently a PhD student in Computer Science at University of Cheikh Anta Diop of Dakar, Senegal. He received his BS in computer Science from University of Cocody (Côte d'Ivoire) in 2003, his M.S. in Numerical Analysis from the Department of Mathematics and Informatics at University of Abobo-Adjamé Abidjan in Côte d'Ivoire in 2005 and professional Master of engineering from CFTIC (Centre de Formation en Technologies de l'Information et de la Communication) at INPHB – Abidjan in Cote d'Ivoire in 2007. His research interests include IPv6 Mobility, IP Multicast, IP network security.