Secure Distributed AAA with Domain and User Reputation

Dan Forsberg
Nokia Research Center
dan.forsberg@nokia.com

Abstract

We explore the problem space of distributing AAA and describe a novel distributed AAA model based on cost efficient hardware security with Access Point specific and domain specific certificates. We introduce new domain and user reputation mechanism on top of the distributed AAA system based on the resources and/or services provided and consumed. Our certificate model includes different provisioning scenarios for scalable deployment and new ways of handling user profiles securely with the help of overlay file sharing networks. Our starting point is similar to community based networks like FON.

1. Introduction

User and data authentication is required if the user pays for the network access in order to prevent service theft. There may be also other reasons to have network access authentication, like a possibility to trace misbehaving users and unwanted traffic (e.g. DDoS, spam, etc.). These authentication, authorization and accounting (AAA) procedures are highly critical and normally centralized, which implies certain requirements for them, like scalability, fault tolerance, and fast response times. AAA servers hold user identity and credential databases and process accounting records for charging purposes. Real-time connection to the user's home AAA server is required in the case of network access authentication and authorization. These kinds of AAA systems cost a lot.

We believe that in order for community based networks like FON [1] and SparkNet [2] to succeed, the costs must be minimized and heavy AAA infrastructures avoided. We realize and assume that APs are on and online most of the time, which is usually not the case of energy inefficient PCs.

- In this paper we explore the challenges of AAA distribution in terms of requirements and problem partitioning.
- We improve the AAA scalability and cost efficiency by distributing it to the edges of the network, to the Access Points (AP), with the help of different certificate deployment scenarios and hardware based security.
- Then we describe how the resulting system could benefit of a new distributed community and user reputation model, protected with the hardware based security on the APs.

We describe the reference architecture, high-level AAA system requirements, motivations for distributed AAA, and evaluation criteria for the solutions (section 2). We explore related work and describe our solution for distributed AAA based on certificates and hardware based security mechanisms [3] (section 3). In the end we conclude our paper and list issues for further study (section 4).

2. Scalable and cost efficient AAA

We are after an architecture where each Access Point (AP) or Access Router acts both as AAA client and server. This architecture is described in Figure 1. All the APs are connected to the Internet and there may be one or multiple APs in a domain (e.g. DNS domain). In our network model each AP can communicate with any other AP. This, however, inherits a clear problem of how can APs have a SA with all other APs.

2.1 Distributed AAA and general requirements for a AAA protocol system

From "Criteria for Evaluating AAA Protocols for Network Access" [4] document we find out different requirements for an AAA systems in general. The first requirement is *scalability* in terms of number of subscribers and the capability of supporting multiple requests simultaneously. The next requirement is *fail-over* in terms of AAA server communication problems. If the default AAA server is unreachable, there must be a backup server handling the requests. Third

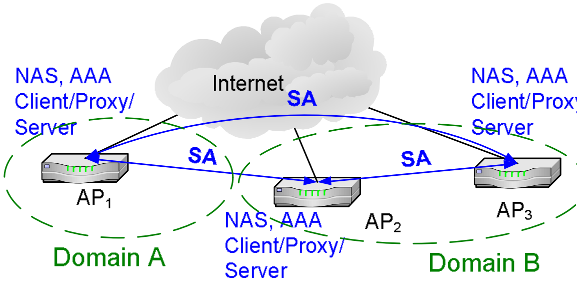

Figure 1. Fully distributed AAA architecture

requirement is about *mutual authentication* between AAA client and the server. This means that both entities need to verity the other end's identity. Fourth and fifth requirements are related to security associations between network elements. In our network model, we use end-to-end security. The ninth requirement is about routing of the AAA packets through *transparent proxies*. This requirement, however, without end-to-end security poses security risks as the proxies can see the packets in plain text (but do not modify them). Tenth requirement is *auditable process* for the packets traversing between the AAA server and an AAA client. In our network model, we think that these are not problematic as the packets are secured end-to-end between AAA client and AAA server. Eleventh requirement is to *allow* using either AAA protocol level security or *underlying protocol security* (like IPSec or TLS). Twelfth requirement is extensibility for the protocol.

As a summary, a distributed AAA system, as we describe it, fulfills the general AAA system requirements.

2.2 Splitting the problem of AAA distribution

We need to address the motivation for the distribution as well as the problems related to mutual authentication between client and AAA server.

In general the AAA server must be scalable. This means that the AAA server should be able to serve as high number of clients as possible. One way to achieve this is to distribute the needed AAA processing. Examples of successful partial localization are the 2G and 3G authentication systems [5], where the home AAA (HLR) server provides triplets (GSM) or quintets (UMTS) for the visited network AAA server, which can then use them to authenticate the user, not only one time, but multiple times, depending on how many triplets/quintets the home AAA provided for the visited network AAA server. In our case this would mean that an AAA server provides something similar for the AAA client. Thus, the problem is also about *(1) How to efficiently decentralize an AAA session, not only the architecture*? This can be achieved for example by partial localization of the authentication protocol. Localization ensures timely delivery of the authentication result.

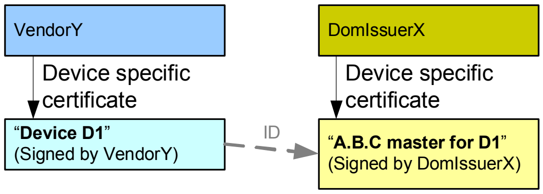

Figure 2. Certificate hierarchy

It is not enough to just think about the protocol, fail over, robustness, scalability etc. requirements, but also overall system requirements in terms of associated usage models. For example, Simon, a subscriber of domain B, has a user profile in the corresponding domain B's AAA server. In the light of the fail-over requirement, the AAA server in domain B should have a backup AAA server in case the main AAA server breaks. However, this would require having at least two (or more) servers per domain, which is not what we want from our distributed AAA system (increased cost). This problem can be described as *(2) How to assure AAA system availability from a single user point of view*?

In case the user's profile is distributed to multiple AAA servers, updating it becomes problem. *(3) Who is allowed to update the user profile and how to ensure its integrity*?

The roaming of users can be stated as a *(4) How to distribute the AAA server for roaming users* problem, as it requires the visited network's AAA server to authenticate the user and act as a AAA client towards the user's home AAA server.

Roaming brings us further to the problem of trust and mutual authentication between the AAA servers and clients. *(5) How can two AAA servers authenticate each other if there is no pre-established security association*?

Establishing initial security associations between AAA servers requires trust and AAA server's identity verification. *(6) How to prevent AAA server identity spoofing.* The question is then also about *(7) Centralized or distributed trust management.* In case the AAA system uses certificates, trust based on the certificates is based on the shared knowledge of the Certificate Authorities. If the system uses some kind of a reputation mechanism, the trust is fuzzy. The PGP system is an example of this. Certificates may also require a revocation system.

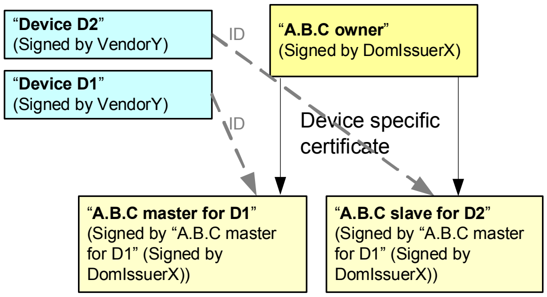

Figure 3. Model-1: Domain owner creates master and slave AAA certificates

Legal interception may be required from the AAA system and also user location tracking. Thus the problem is also that *(8) How to allow and implement user location and traffic protection keys queries for authorized parties only*.

Finally the system must also be able to issue client credentials for new users and also for user's who have lost their currently active credentials. Thus, the problem is also about *(9) How to do user account and credential management and accounting with control possibilities for authorized parties only*.

3. Distributed AAA

Our starting point for the distributed AAA architecture is shown in Figure 1. In this section we describe our four solution building blocks.

First, we think that reputation based systems are not solely strong enough to build a distributed AAA system, and that a certificate based mechanism is an obvious candidate for establishing security associations between the AAA servers (e.g. PKI). We need to have a CA and naming structure for **(a) device specific certificates**. However, we also need **domain specific certificates** that provide capability for an AP to authenticate itself as an AAA server for a particular domain. A device specific certificate should be issued during the manufacturing time by the vendor. Domain name issuer provides the domain specific certificate.

Secondly, we think that a suitable **(b) hardware based security building block**, like a Trusted Platform Module (TPM) [3] chip, is an evident and cost efficient evolution step for improving AP security. TPM is used to protect the secret keys of the corresponding public key certificates and the AP software integrity. Certificates are used to authenticate the AAA servers, when they communicate to each other.

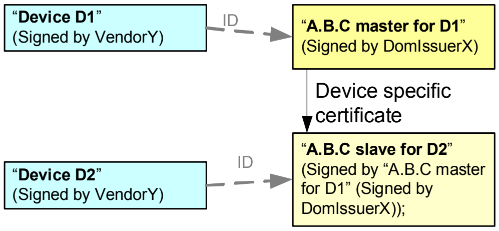

Figure 4. Model-2: Domain owner or device can create the slave AAA certificate

Thirdly, we need to provide redundancy for fulfilling the fail-over requirement. For this purpose we need a **(b) domain backup or secondary AAA server discovery and assignment mechanism**. This procedure, as any other procedure, should be automated as much as possible to lower the management overhead. AAA Diameter protocol specification describes how to use DNS to discover one or multiple domain AAA servers. The DNS NAPTR [6] can be used to discover the slave (e.g. secondary) AAA servers.

As a fourth building block, we need to take care of the user profile handling. We can use an **(4) overlay network to store encrypted user profile data** for the domain it belongs to.

3.1 Device and domain certificate management models

We describe the device and domain specific certificate management and the 4 different models for automatically and securely assigning an authorized domain slave (e.g. secondary or backup) AAA server.

Vendor Y creates a signed device specific certificate. This certificate is securely stored into the AP with the help of the TPM chip. These device specific certificates (possibly along with remote attestation) are used to build the SAs between the AAA servers.

A domain issuer provides certificates for users (or organizations etc.) that purchase a new domain name. This domain specific certificate is bound to a device identity so that only this specific device can act as a domain master AAA server (see Figure 2).

From the authorization and authentication point of view the domain slave AAA server delegation requires further certificate hierarchy management. There are different varying models on how this could be done.

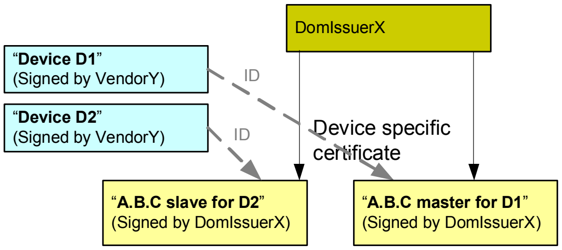

Figure 6. Model-3: Domain issuer provide master and slave certificates

In the first model (Figure 3) the **(1) domain issuer provides a master domain certificate to the user, which is NOT bound to any device id**. This way the user can freely choose what devices it wants to use for domain master and domain slave servers. Attacker getting hold to this certificate would then be able to create both domain master and domain slave certificates. The domain owner certificate is a critical asset for the domain and must be protected well. Suits best for medium and big setups of centralized operations and management servers.

In the second model the **(2) device bound domain master certificate is used to sign and bind a domain slave certificate with a device id** ("D2" in Figure 4). This model is the most allowing for the domain slave certificate creation. This model allows both the user and the device holding the master domain certificate to create slave domain certificates. Thus, an attacker getting hold to the master domain certificate can create domain slave certificates.

Third model is the most stringent (Figure 5). In this model the **(3) domain issuer provides both domain master and domain slave certificates, both bound to a device id**. The model does not allow automatic domain slave server nomination as the domain master server can not create the proper certificate.

In the last model (Figure 6), the **(4) device holding the domain master certificate creates domain slave certificate(s)**. The domain slave certificate must be signed by the domain master device's certificate. This model is best suited for our distributed AAA system, because the domain slave certificate can only be created by the domain master server. Attacker getting hold to the domain master certificate does not get any benefit as the device is needed to create the certificate (signature with the device specific certificate).

3.2 Rating system

Creating a distributed AAA system with *domain and user rating* system (see Table 1) is possible because of our assumption of having software integrity (based on secure hardware) with device and domain

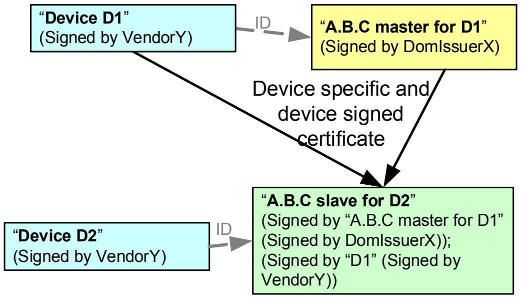

Figure 5. Model-4: Both domain and device must sign the slave certificate

certificates. Otherwise the system would be easy to compromise and thus building a distributed reputation system would not be feasible. Note that the domain and user rating is not used to maintain or establish SAs between AAA servers.

From the distributed AAA system point of view *domain rating* increases if the domain APs are online time is higher and/or if the coverage is extended by installing new APs. Rating increases also by providing computational resources and storage capacity for the distributed system. Increasing the number of domains that a domain can serve and higher guaranteed bandwidth share available for others also increases rating of the domains.

User rating increases if she instructs somebody else to join the community and to install an AP. Rating gets lower when the user consumes more storage capacity on the overlay network than she is currently offering. Rating also decreases if the user does not keep her AP online all the time. If the rating is low for the domain, the number of possible user accounts is lower as well.

3.3 User profile handling

The user profile may contain static and dynamic parameters. For the static data (e.g. is not updated very often) overlay network storage can be used (see Figure 7). The user profile data must be protected from being modified or deleted by malicious nodes and/or users. Thus, the user's home domain AAA server integrity protects the user profile data and authenticates profile modifications[1].

For user profile confidentiality protection the client negotiates with the home domain master AAA server a key that is used to encrypt and decrypt the user profile data. The user profile is decrypted only if the user and AP achieve mutual authentication (client provides the key). This is possible only if the home domain AAA server has authenticated the client's AP.

[1] Preventing user profile deletion from the overlay network is out of the scope of this paper

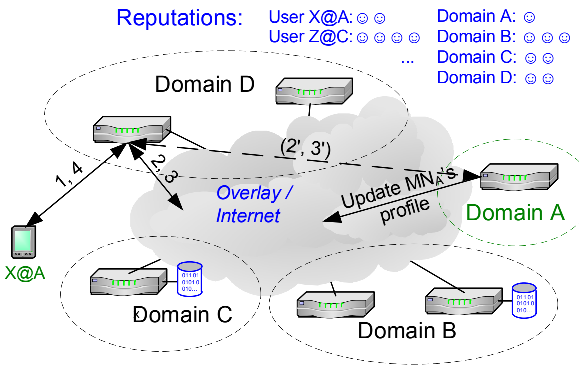

Figure 7: User profile retrieval from the overlay or from the home domain

For dynamic data like for example client authentication keys for APs, accounting records, and etc. static user profile can not be used. For this purpose the home domain AAA server is contacted. However, for pure authentication and authorization reasons it is not necessary to contact the home domain AAA server every time if AP dedicated keys can be found from the overlay network. For this purpose the home AAA server creates a Session Key Context (SKC) [7]. With this method it is also possible to store the user profile encryption key into the dynamic user profile data.

In case there is no previous SA with the home domain AAA server or there is no data for the user in the overlay network, the home domain AAA server must be contacted.

Checking if user profile data is available on the overlay network can be time consuming especially if the data is not present. Thus, the client could indicate whether the data exist or not and optionally provide an index for the user profile data (see Figure 8). However, based on user mobility patterns, the APs could store the recent users' user profile and service related data to speed up the user profile and user specific service related data from the overlay network. This way the distributed AAA network adapts to the user's mobility patterns and improves service quality (faster profile access).

4. Related work

Liang and Wang [8] describe how localized AAA control scheme improves the performance of the AAA system. The local AAA server creates a SA with the visiting clients so that the clients can be authenticated multiple times in the domain without necessarily involving the client's home domain.

Zrelli and Shinoda [9] describe in their paper a protocol that can be used to extend Kerberos to support authentications across different domains. In essence they allow the home Kerberos server to deliver user specific credentials to the visited domain, which can then use them to authenticate the user and provide credentials for service access. This way only one authentication protocol run with the home domain is needed for both the network and the service access.

Table 1: Rating attributes

Attribute	User	Domain
AP uptime (%)	+	+ (total)
Domain coverage (APs)		+
Connection capacity	+	+ (total)
AP storage capacity	+	+ (total)
Storage consumption	-	- (total)
Computational resources	+	+ (total)
Serving other domains		+
Guaranteed bw for visitors		+
Bring new users	+	
Users' reputation (behavior)	+	+
Services/content sharing	+	+ (total)

Ngai and Lyu [10] propose a scalable, secure, and distributed authentication service that enhances the correctness of public key certification in wireless ad-hoc networks in the presence of malicious nodes. In their schemes the authentication is based on public key certificates, distributed CA functionality, and CA reputation. They also cover other related work in the area of distributed CA.

5. Conclusion

Decentralizing AAA has multiple problems, mainly related to user account integrity, user profile management, fail-over mechanisms, and AAA interworking for roaming users. We explored this problem space and described our distributed AAA system based on hardware security and certificates with multiple deployment models. We required common CAs, which could for example be the AP vendors. We described how a novel, distributed, and two layer (domain and user) rating system can be used with our distributed AAA system to build incentives for users to maintain, contribute, and collaborate in the network. Our system is flexible and scales from single AP domain to multiple AP domains. User account management can also vary based on the domain policy and the domain reputation.

Our distributed AAA system requires that all the APs are running non-malicious software and that by using mechanisms standardized by TCG (for example secure boot and remote attestation) the trust can be distributed among the APs with good enough security.

The distributed AAA model with user and domain rating can be used as a platform for launching new services without high investment costs and risks.

Next steps include refinement of the rating model and test bed implementations. Security analysis, proofs, and detailed implementation requirements are left for further study.

Acknowledgements

We would like to thank Mikael Latvala, Seamus Moloney, and Sateesh Kavuri for interesting discussions and good comments.

References

[1] FON, Large WiFi Community and Social Router, URL: http://www.fon.com/ (referenced 2007-04-13)

[2] SparkNet and OpenSpark – Innovative wireless local area network (WLAN), URL: http://www.sparknet.fi/ (referenced 2007-04-13)

[3] Trusted Computing Group, URL: https://www.trustedcomputinggroup.org/ (referenced 2007-04-13)

[4] Aboba, B. et. al., "Criteria for Evaluating AAA Protocols for Network Access", IETF RFC 2989, November 2000.

[5] 3GPP TS 33.102: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; 3G Security Architecture".

[6] Mealling, M. and R. Daniel, "The naming authority pointer (NAPTR) DNS resource record," IETF RFC 2915, September 2000.

[7] D. Forsberg, "Protected Session Keys Context for Distributed Session Key Management", Springer's International Journal on Wireless Personal Communication, special issue on Seamless Handover in next Generation Wireless/Mobile Networks., February 2007 (to be published).

[8] Wei Liang, Wenye Wang. "A Local Authentication Control Scheme Based on AAA Architecture in Wireless Networks" In Proc. 2004 IEEE Internation Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, June 2004. IEEE Computer Society Press.

[9] Edith C.H. Ngai, Michael R. Lyu. "An Authentication Service Based on Trust and Clustering in Wireless Ad Hoc Networks: Description and Security Evaluation" In Proc. 2006 IEEE Internation Conference on Sensor Networks, Ubiquiotous, and Trustworthy Computing, Taichung, Taiwan Conference, June 2006. IEEE Computer Society Press.

[10] Saber Zrelli, Yoichi Shinoda. "Single sign-on framework for AAA operations within commercial mobile networks" In Proc. 2006 IEEE International Conference on Availability, Reliability and Security, Vienna, Austria, April 2006. IEEE Computer Society Press.