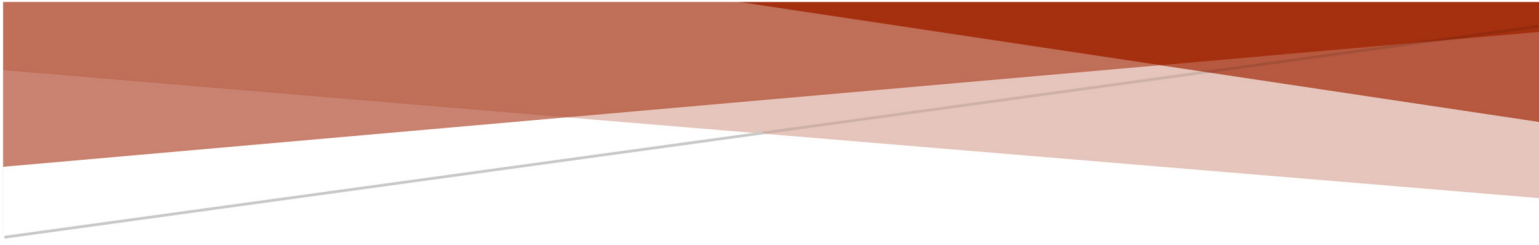


METHOD FOR MAC ADDRESS UNIQUENESS CHECKING  
VIA HIERARCHICAL RANGES

INVENTOR:

MASSIMILIANO PALA



# METHOD FOR MAC ADDRESS UNIQUENESS CHECKING VIA HIERARCHICAL RANGES

Range-Based MAC uniqueness checking

## Abstract

This work describes a novel approach to checking for uniqueness of MAC addresses

Massimiliano Pala  
m.pala@cablelabs.com

<b>1</b>	<b>OVERVIEW</b> .....	<b>3</b>
1.1	CURRENT APPROACH AND PROBLEM STATEMENT .....	3
<b>2</b>	<b>THE INVENTION</b> .....	<b>3</b>
2.1	ENTITIES AND THEIR FUNCTIONS.....	3
2.2	ALLOCATING MAC ADDRESSES .....	3
2.3	CHECKING FOR MAC ADDRESSES UTILIZATION .....	5
<b>3</b>	<b>HIERARCHICAL SUPPORT FOR SUB-RANGE CHECKING AND MANAGEMENT</b> .....	<b>6</b>
3.1	HIERARCHICAL DEPLOYMENT EXAMPLE: DOCSIS PKI .....	6
3.1.1	<i>Pre-Allocating Customers' MAC ranges</i> .....	6
3.1.2	<i>Updating Customers' MAC ranges</i> .....	6
3.1.3	<i>Checking sub-allocations of MAC addresses</i> .....	7

## Bibliography

- [1] The Internet Engineering Task Force (IETF) – IETF RFC 5280. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, edited by W. Polk et al., May 2008. Also available at <https://datatracker.ietf.org/doc/rfc5280/>
- [2] The Internet Engineering Task Force (IETF) – IETF RFC 2987. *PKCS#10: Certification Request Syntax Specification Version 1.7*, edited by M. Nystrom et al., November 2000. Also available at <https://datatracker.ietf.org/doc/rfc2986/>

# 1 Overview

Many protocols use MAC addresses as identifiers for devices - devices with different MAC addresses are considered separate entities. In the PKI world, CAs may be required to check for uniqueness of values – i.e., checking if a MAC address was already used in a different certificate. These checks can take quite some time because of the large dimension of the search space (48-bits).

This invention reduces the lookup procedures requirements by leveraging the concept of constrained ranges of addresses.

## 1.1 Current Approach and Problem Statement

Searching for the existence of a value among a large set of values may require a lot of time and not be practical because of the size of the search space. We remove these issues by using the concept of ranges instead of individual checks.

# 2 The Invention

The invention describes the procedures for how to handle (store and update) MAC address ranges and the associated lookup mechanism for duplicate values.

## 2.1 Entities and their functions

The following entities are defined:

- **The relying party.** This is an application or entity that is checking if a MAC address or a range of MAC addresses are still available (not already used)
- **The MAC addresses to be checked.** This is the set of MAC addresses or MAC addresses ranges that need to be checked for uniqueness. This entity can also be responsible to update the available MAC addresses ranges or this function can be associated with a separate entity.
- **MAC Address range.** This is a range of MAC address and it is expressed as an initial and a final value for the range.
- **The Available MAC addresses.** This is the set of MAC addresses or MAC addresses ranges that are currently available for allocation. This set changes with each allocation as the number of available addresses shrinks. This set can comprise the entire address space or a subset of that. Different entities might be assigned different pre-allocated Available MAC addresses.

## 2.2 Allocating MAC addresses

When new MAC addresses are allocated / used, their values must be removed from the Available MAC addresses. Specifically, for an allocation of values  $\{ X_m, X_n \}$ , there are three different cases:

- The range is fully enclosed in a continuous available MAC addresses range. In this case, the enclosing range is split into two different ones where the newly assigned values are removed:

$$\{ X_1, X_Z \} \rightarrow \{ X_1, X_{m-1} \}; \{ X_{n+1}, X_Z \}$$

- The range exceed the upper limit of the available MAC addresses range. In this case, the end of the available MAC addresses is updated to exclude values higher than  $X_m$  (i.e.,  $\{ X_m, \dots \}$ ):

$$\{ X_1, X_Z \} \rightarrow \{ X_1, X_{m-1} \}$$

- The range starts with a value lower than the lower limit of the available MAC addresses range. In this case, the beginning of the available MAC addresses range is updated to exclude values lower than  $X_m$ :

$$\{ X_1; X_Z \} \rightarrow \{ X_{m-1}; X_Z \}$$

The allocation algorithm is as follows:

Input:

$R_x$  - Range to be checked.  $R_x^{INIT}$  is the start of the range.

$R_x^{END}$  is the end of the range.

AMAR - Set of ranges for available values

Output:

Updated set of ranges for available values.

Algorithm:

For aRange in AMAR:

// Checks the beginning of the range

if ( $R_x^{INIT} < aRange^{END}$ ) next;

// Checks the end of the range

if ( $R_x^{END} > aRange^{INIT}$ ) next;

// Removes the first elements from the range

if ( $R_x^{INIT} \leq aRange^{INIT}$ )

$aRange^{INIT} = R_x^{END} + 1;$

else

    if ( $R_x^{END} \geq aRange^{END}$ )

$aRange^{END} = R_x^{INIT} - 1;$

    else

        Allocate newRange;

$newRange^{END} = R_x^{INIT} - 1;$

$newRange^{INIT} = aRange^{END} + 1;$

$aRange^{END} = R_x^{INIT} - 1;$

## 2.3 Checking for MAC addresses utilization

When checking if a range (can be a single value) has already been used, the relying party actually checks if the requested range is all enclosed in one or more available MAC address range(s).

The Process starts from the initial ranges of available MAC addresses. These values can cover the whole address space or just a subset.

When checking if a MAC address or a MAC address range was previously allocated/used, the values of the available MAC addresses ranges are checked. Specifically, if the range values to be checked are included in any of the available MAC address ranges (i.e., the beginning is after the initial value of the available MAC addresses range and the end is before the end of the available MAC addresses range) then the range has not been allocated/used yet. On the contrary, if no available MAC addresses ranges completely include the requested range, the range is not available for allocation.

The lookup algorithm is as follows:

Input:

Rx - Range to be checked.  $R_x^{INIT}$  is the start of the range.  
 $R_x^{END}$  is the end of the range.  
AMAR - Set of ranges for available values

Output:

True if Rx is fully contained, False otherwise.

Algorithm:

```
For aRange in AMAR:
    // Checks the beginning of the range
    if ( $R_x^{INIT} < aRange^{INIT}$ )
        next;
    else
        // Checks the end of the range
        if ( $R_x^{END} > aRange^{END}$ )
            next;
        else
            // Range is fully enclosed
            return True;

return False;
```

## 3 Hierarchical Support for Sub-Range Checking and Management

In order to provide better management, the allocation and validation can be applied to the whole address space or to subsets of it. The allocation can be structured hierarchically where allocations at different level of granularity are applied.

For example, one entity can be set to manage a global space allocation – for example, ranges of MAC addresses can be pre-associated with specific entities/customers thus keeping track of allocation at a per entity/customer level. A second level in the hierarchy can be associated with each entity/customer with a pre-assigned range or set of ranges. In this case, checks are performed against the assigned set of ranges thus reducing the search space considerably. A third level in the hierarchy could be managed to pre-allocate MAC addresses to different sub-entities/departments thus guaranteeing that the different sub-entities/departments do not request duplicate MAC addresses

### 3.1 Hierarchical Deployment Example: DOCSIS PKI

In the DOCSIS environment, MAC addresses are used to tie a keypair to a hardware address (MAC Address). This invention can be used to make sure that assigned MAC addresses are not repeated in certificates issued to the same or different customers.

#### 3.1.1 Pre-Allocating Customers' MAC ranges

MAC addresses are usually allocated by IEEE in ranges. The values in those ranges are then used to identify the different devices. This invention enables the possibility to hierarchically manage sub-allocations.

Specifically, a set of allocations from the general address space is performed by a central authority who is in charge of allocating ranges for the ecosystem.

When new participants to the ecosystem are added (e.g., new customers) the requested (or assigned) ranges of MAC addresses is checked against the global allocation and rejected if the range is not available (e.g., already fully or in part allocated for another customer).

Customers are allowed to use only the values within the pre-allocated range and track the internal allocation of addresses within the assigned range.

#### 3.1.2 Updating Customers' MAC ranges

When the initial allocation for the customer MAC addresses is insufficient, the allocated initial range can be augmented by extending its range or by adding new ranges to it. Exactly for the case where new customers/entities are added to the ecosystem, when augmenting with additional ranges (or extending the current one), the requested new ranges are checked against the global space (previous level in the hierarchy) – if they are available, the request is granted and both the global space allocation ranges and the assigned range to the entity are updated accordingly.



### **3.1.3 Checking sub-allocations of MAC addresses**

In the DOCSIS environment, the customers are pre-authorized to use a subset of the allowed space according to domain-specific rules. When the customer requests a new set of certificates, the requested MAC addresses is checked against the available ones for the customer.

The values submitted by the customers are first aggregated in ranges (either by extracting the values from a provided text file or from the certificate requests themselves [2]) and then the checking algorithm is used to verify all the requested ranges are available. If they are not available, the whole batch of requests is rejected.

If the requested ranges are available, when the request is queued, the ranges are updated to remove the pre-allocated set for the request batch from the available set. In case of errors while processing the request batch that causes its rejection, the pre-allocated ranges are released as available again.