Making a Subset-Difference

Understanding the Crypto that enables AACS

Disclaimer

This is based on my understanding of the AACS specification document

I have not had to fully implement it

I have not seen code for an implementation of it

I’m presenting this because I couldn’t find material to learn from.

If you know better or want to correct me,
please feel free to shout up or contact me afterwards

What am I covering?

- Brief description of AACS
- Explanation of the Subset-Difference tree
  - Structuring of the key system
  - How it enables “efficient revocation of non-contiguous sets”
- A reference codebase
- Some things to help you when reading the specification
  - (if I talk too fast)

What is AACS?

- **A**dvanced **A**ccess **C**ontent **S**ystem
- Started 2004, the spec I have is from Oct 2012
- Anti piracy system for the “new” Blu rays and HD DVDs
  - Improvement on the Content Security System (CSS) of DVDs
- Collaboration of a number of big players:
  - Intel, IBM, Microsoft, Panasonic, Sony, Toshiba, The Walt Disney Company, Warner Bros.

Cryptographic Aims of AACS

- Protect media from being accessed by un-licensed devices
- Allow revocation of a device's access
- Revocation of *Stateless* Receivers
- Should not significantly impact space available on the disc
- Coalition resistant:
  - a coalition of all revoked devices cannot access the media

Revocation of Stateless Receivers

- *"Revocation and Tracing Schemes for Stateless Receivers"* Dalit Naor, Moni Naor, Jeff Lotspiech.
- We do not assume any device can be remotely updated
  - This accounts for malicious "non-patch" strategies
- Con: We cannot retroactively revoke access
  - Devices will still be able to decrypt any data recorded prior to the revocation

How the media is encrypted in a brief 3 point slide

- Each Title (piece of content) is encrypted with a “Title Key”
  - AES-CBC of the content
- A disc’s Title Keys are encrypted with the “Media Key”
- The Media key can be derived by a device from it’s factory-given set of device keys.
  - Unless it’s been revoked.

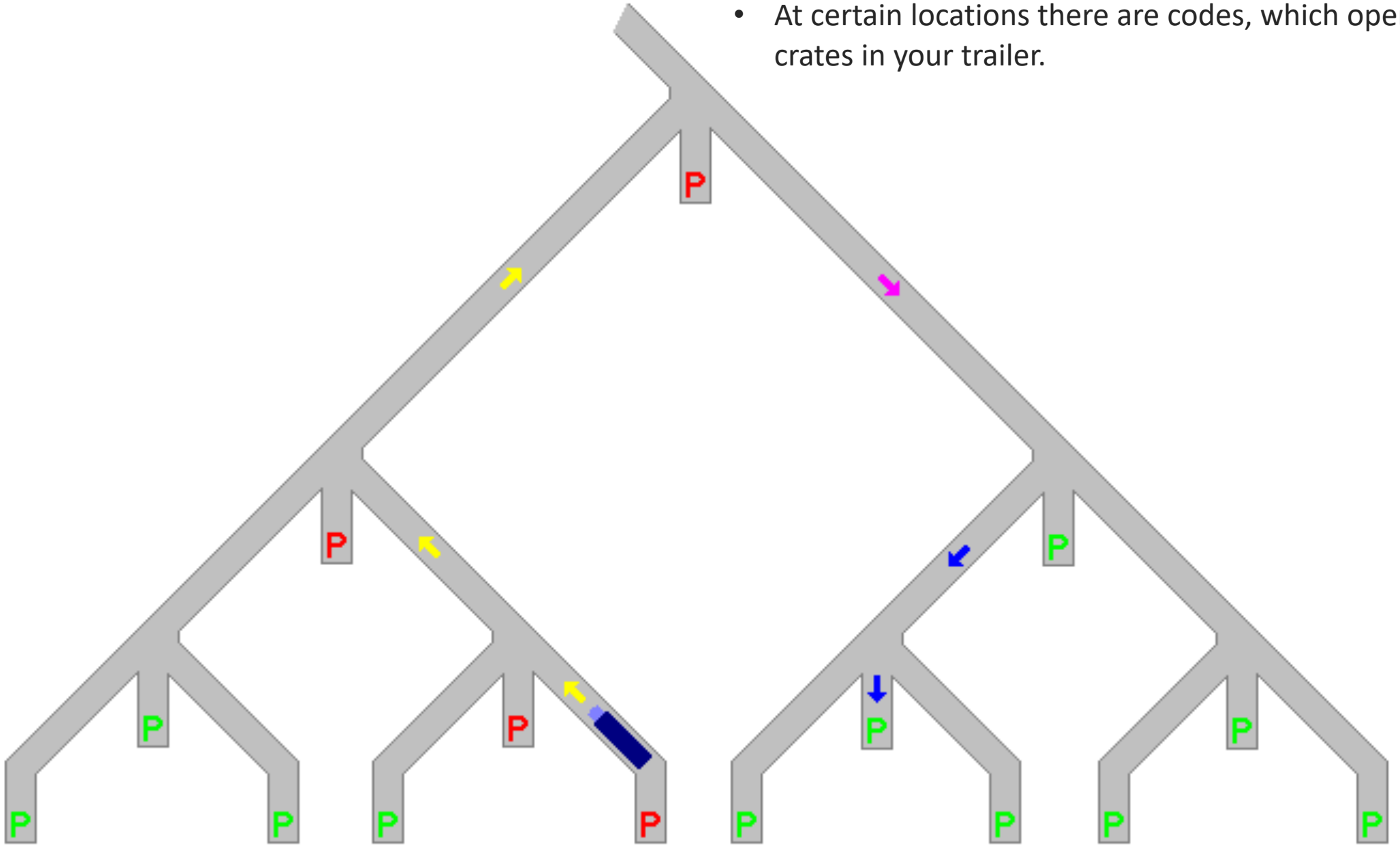
Subset-Difference Tree

The “arnezami” (of Doom9) description: Trucks in a carpark.

- Each Device is a Truck with a trailer, in a crazy car park/road system
- A given truck has a fixed starting point.
- A truck cannot reverse, nor can it take corners < 90°
- At certain locations there are codes, which open one of the crates in your trailer.

https://web.archive.org/web/20070303012909/https://forum.doom9.org/showthread.php?t=122363

- A truck cannot reverse, nor can it take corners < 90°
- At certain locations there are codes, which open one of the crates in your trailer.

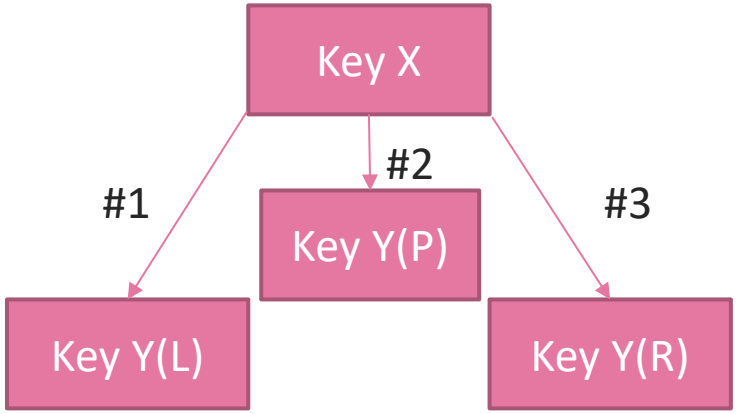


Explanation

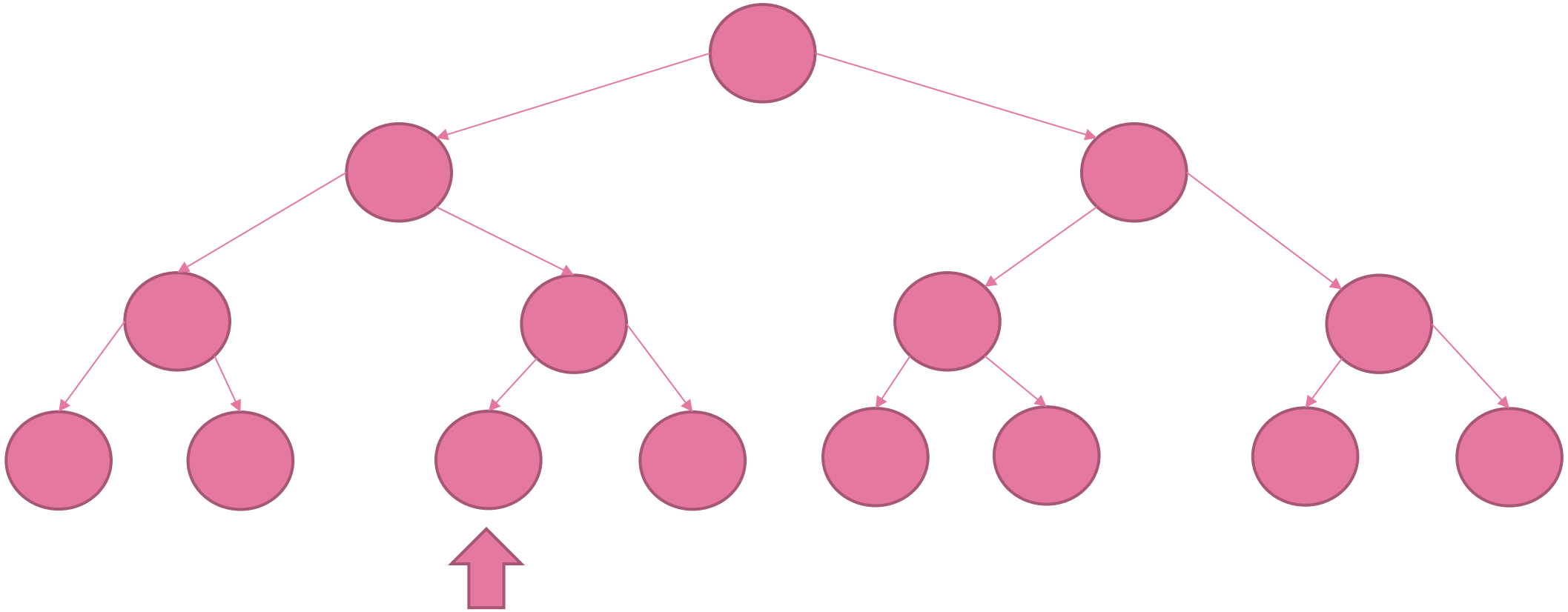
- The Truck is your **Device**
- The crates in your trailer are the (encrypted) **Media Key Data** entries, given to you by the disc.
- The codes at parking locations are **Processing Keys**.
- The prize in the crate is the **Media Key**

How does it work?

- Each set of keys is derived from keys higher in the tree
  - Each node's key is hashed three times, giving the next node to the left, the processing key, and the next node to the right.
- You get every key that's one decision different to the path to your device node
  - Thus you can thus figure out any keys *not* on your path to the root.



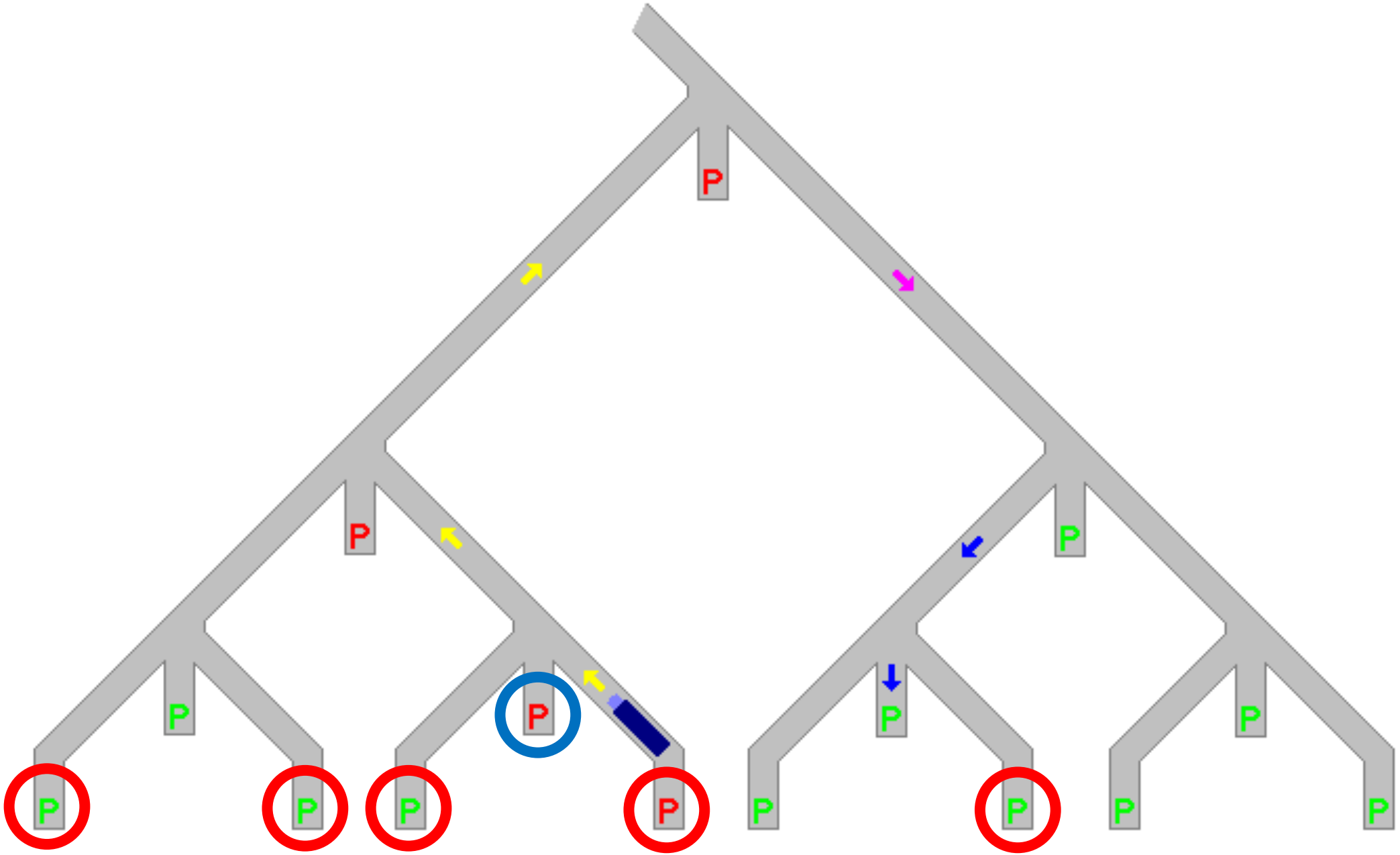
“every key that’s one decision different to my path”?



Revocation

To revoke a node or subset of nodes,

we encode the media key with a processing key that they cannot access


P
P
P
P
P
P
P
P
P
P
P
P
P
P
P

The Subset difference is like ogres
(which are like onions)

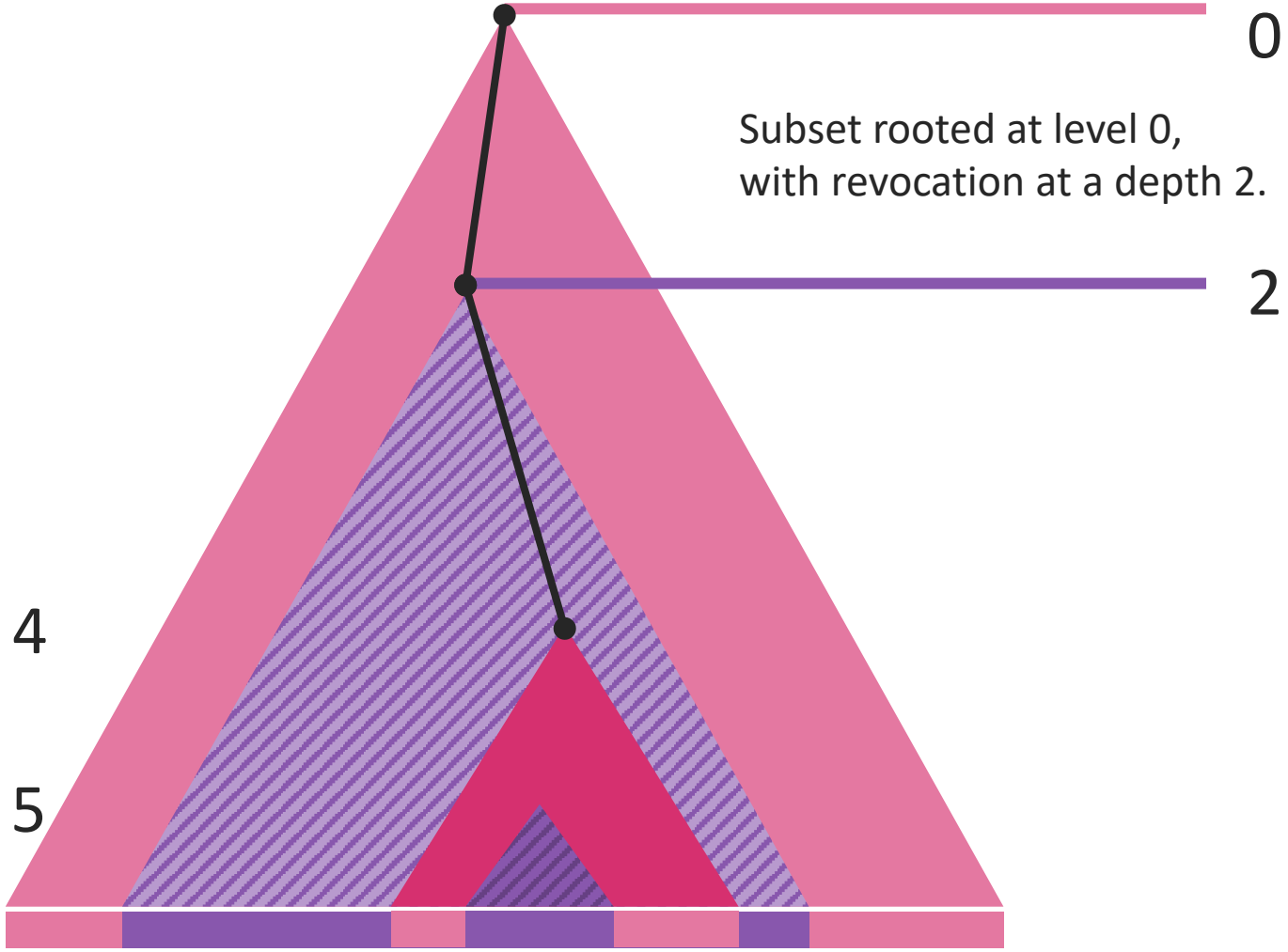
- Discontinuous revocations require layers
- For each level, there is another sub tree of n-1 height

with any other Device Key set can. Further, corresponding to every *sub-tree* in the master tree is another system of keys. For example, one level down from the root of the master tree there are two sub-trees, each with its own system (tree) of keys, in addition to the system of keys in the master tree. Likewise, in the next level down in the master tree there are four sub-trees, each with its own system of keys. By the time you reach the bottom of the tree, each pair of devices belongs to their own sub-tree of height 1. For each sub-tree corresponding to a node in the master tree

How does this help?

- We can do one encryption in each tree, which will revoke a subset of nodes
- In a lower set, we can re-enable some nodes and revoke a smaller subset.

Revocation in layers



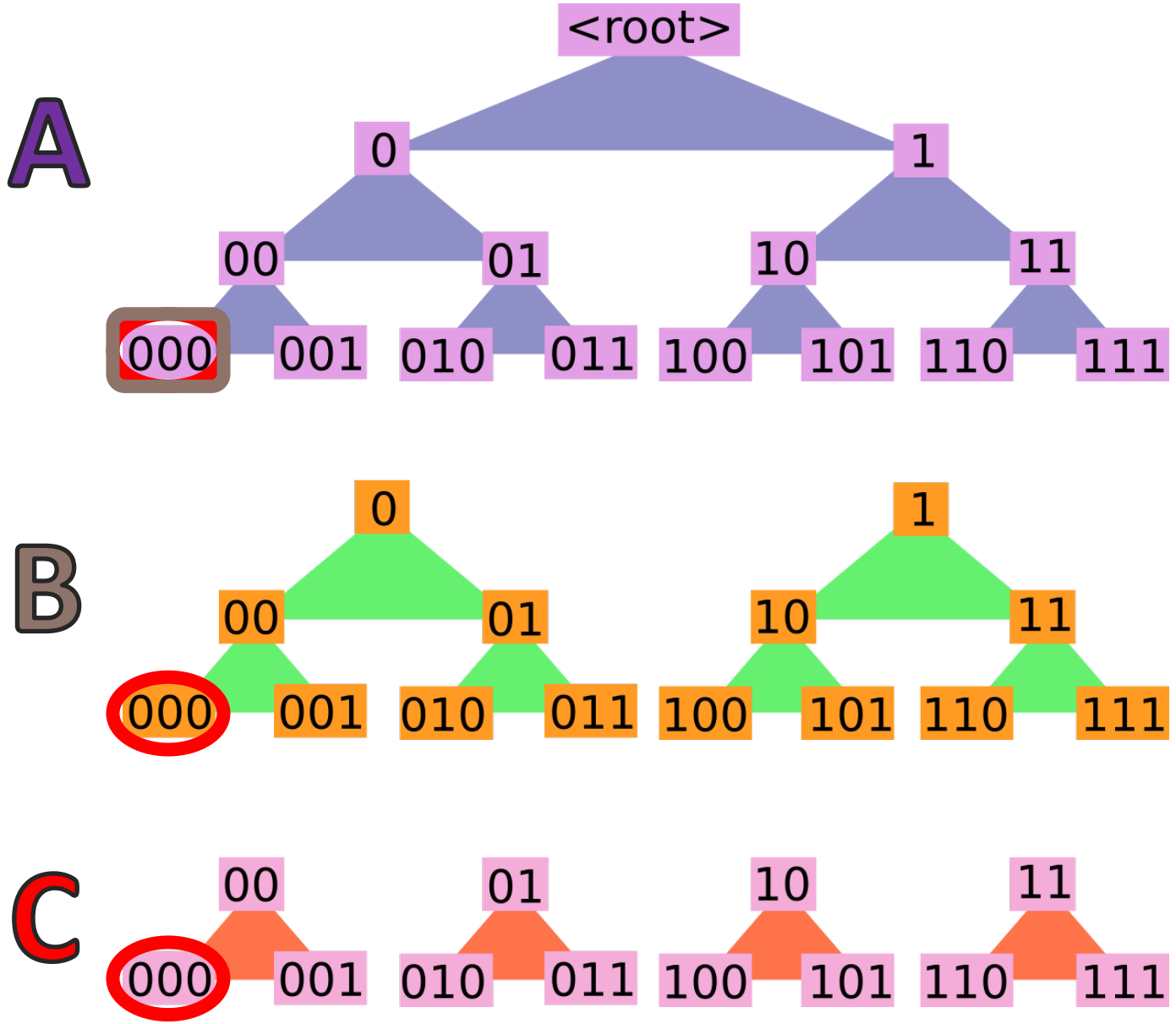
Example Revocations

Revoke:

- **000**

Encrypt with:

- **A-000**



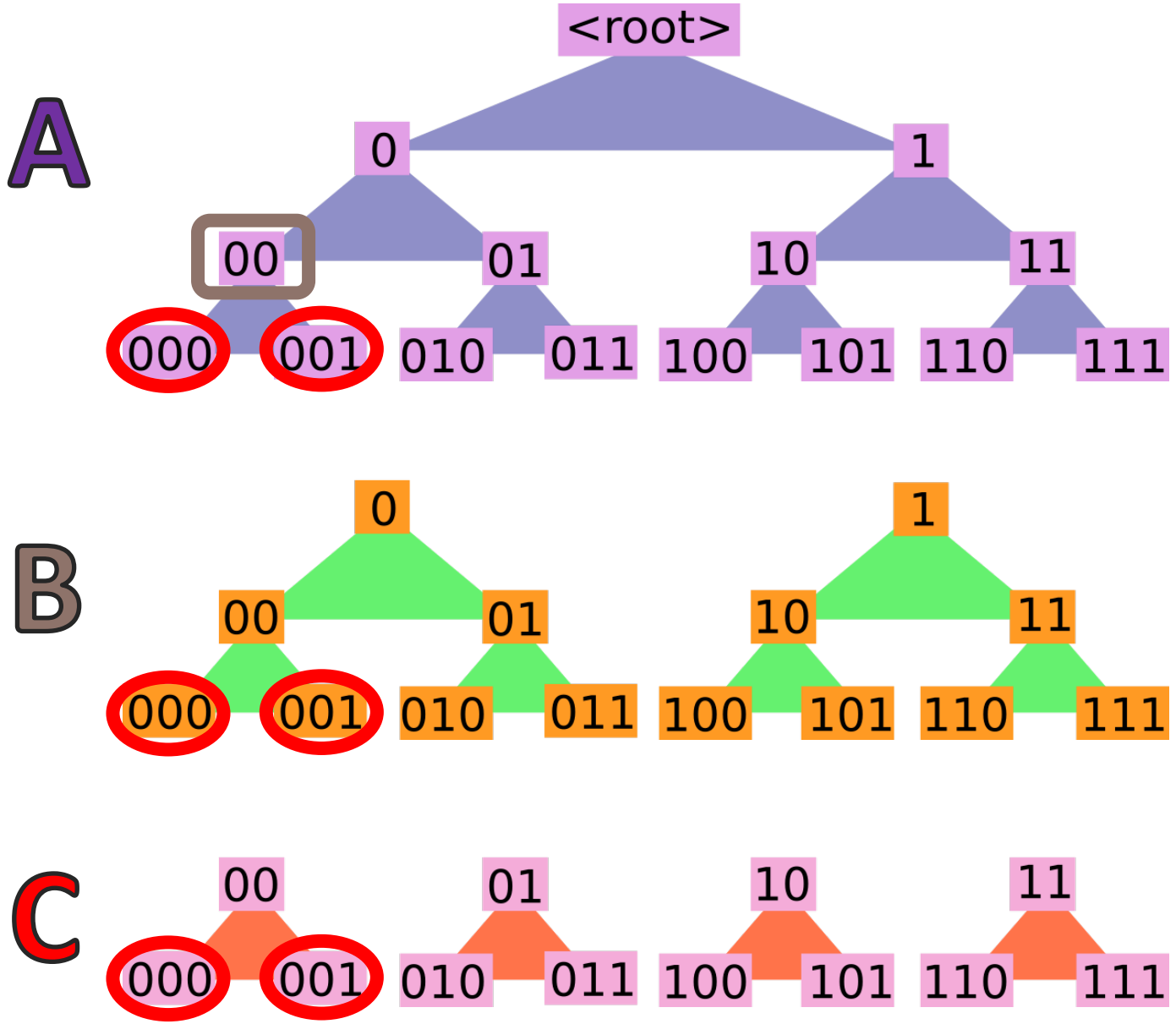
Example Revocations

Revoke:

- **000**
- **001**

Encrypt with:

- **A-00**



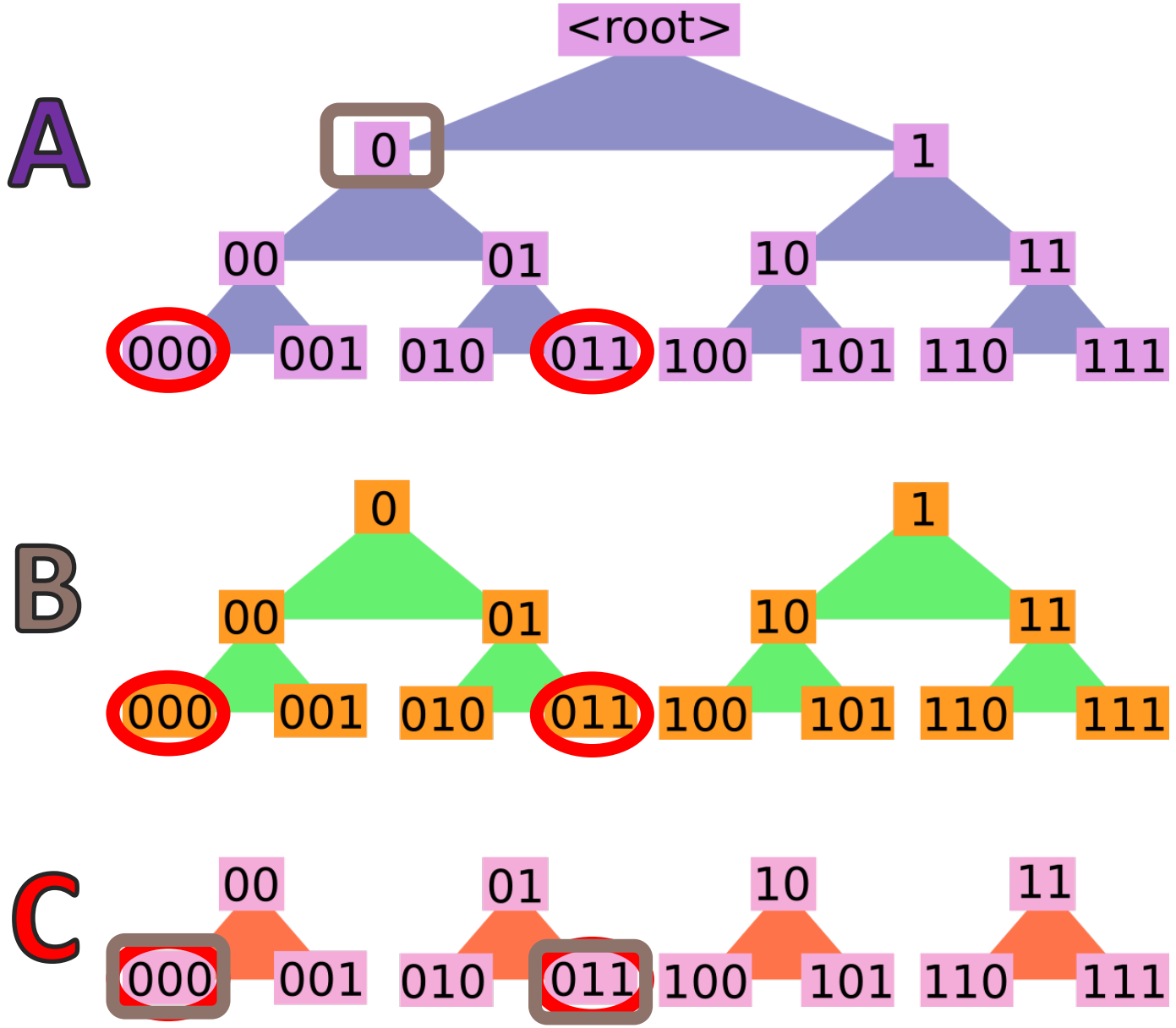
Example Revocations

Revoke:

- **000**
- **011**

Encrypt with:

- **A-0**
- **C-000**
- **C-011**



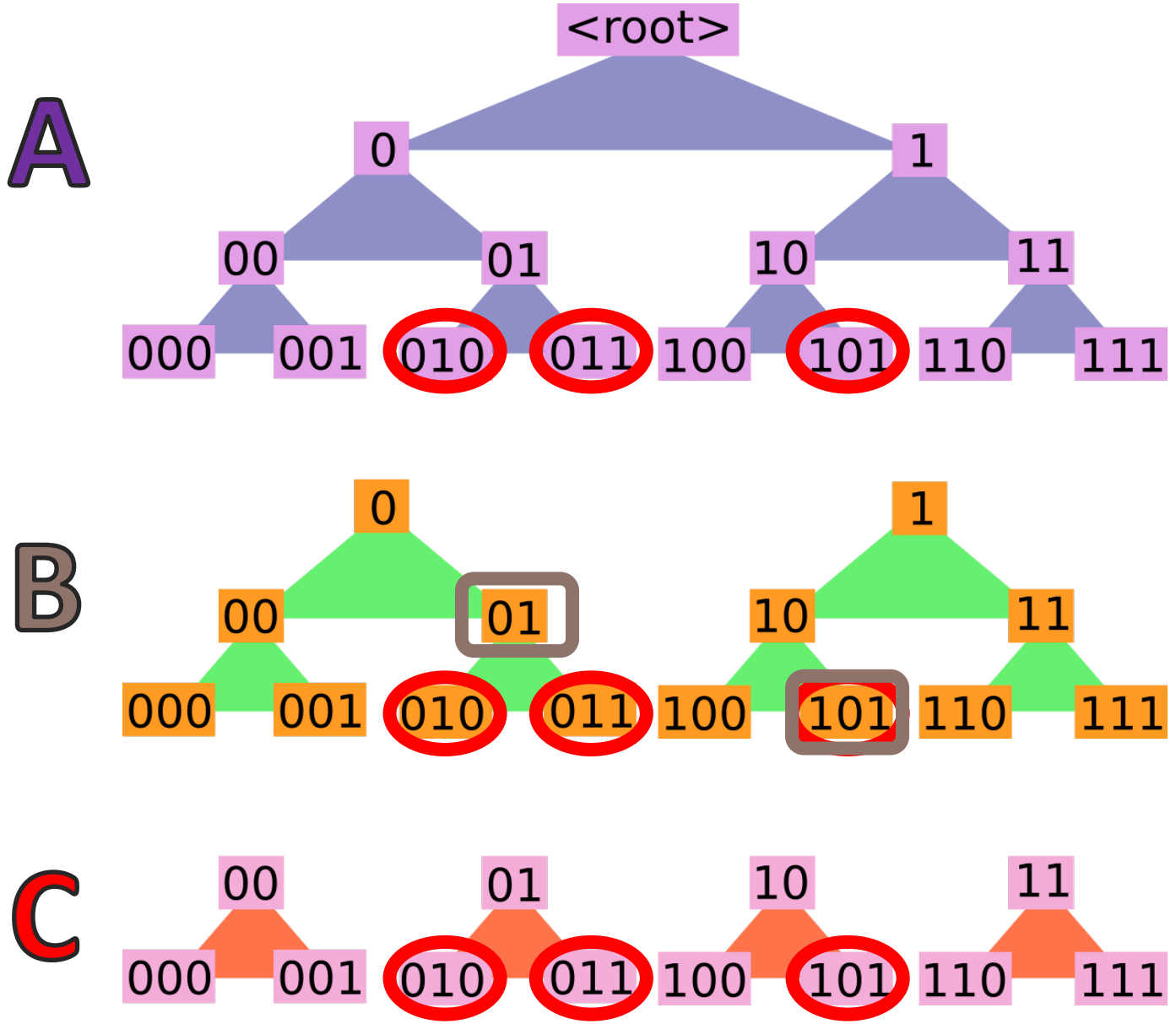
Example Revocations

Revoke:

- 000
- 011
- 101

Encrypt with:

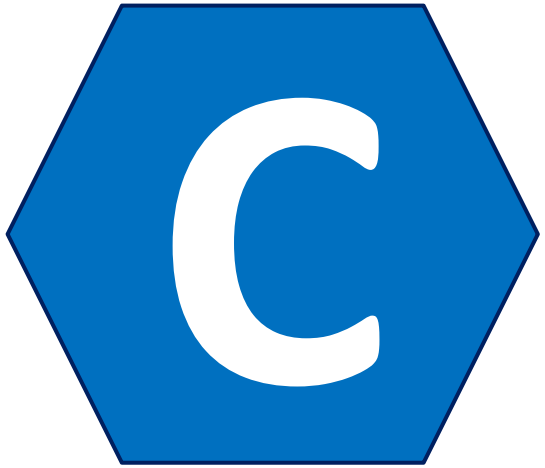
- B-01
- B-101

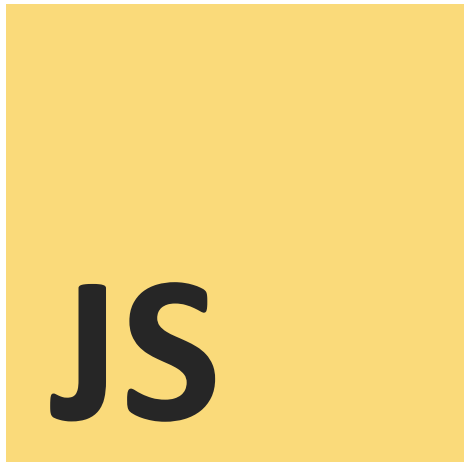


https://github.com/E314c/AACS

A reference implementation of the subset difference parts of the AACS specification.

Please feel free to example implementations in other languages and submit a PR so that more people can read and understand the system





Bonus material

Confusing storage

- Node numbers are stored with an additional ‘1’ bit set on the end
  - To indicate the depth of the node within the master tree
  - Allows you to combine the path and v_mask of a node
    - Referred to as a ‘UV number’
  - The UV number of a device is a “device node number”

The UV Number of node @ ‘1001’ in a master tree of depth 7

1	0	0	1	1	0	0	0

A device node number @ ‘1001100’ in a master tree of depth 7

1	0	0	1	1	0	0	1

Space requirements

- UV numbers are specified as 32 bits, so there's 31 layers of nodes
- You need N keys from a tree of depth N

$$\sum_{N=0}^{N=31} N = 496\ deviceKeys$$

$$496 \times 16Bytes \approx 8kB$$

My node, in the middle of my tree

- Masks ”U” and “V” are used to define valid paths in a subset difference
  - The masks represent which bits must match to be part of that path
  - U is the ancestor path, the mask of U indicates which layer and path the root of this subset is on
  - V is the difference path, the mask of V indicates paths under U that **are not** valid (have been revoked)

I'm wrong or the spec is wrong

. It finds the appropriate stored Device Key as follows: assuming the Explicit Subset-Difference Record value is uv, m_u, and m_v, and the stored Device Key has uv', m'_u, and m'_v, the appropriate Device Key is the one that meets the following condition:

$$(m_u == m'_u) \text{ and } ((uv \,\&\, m'_v) == (uv' \,\&\, m'_v))$$

Verifies it's the same sub-tree depth

Attempts to check that upper part of key and subset match. (Key can derive node under it)

Problem:

- Key path: 01
  - UV = 0110
- Subset exclusion: 0
  - UV = 0100

$$(0000 == 0000) \,\&\&\, ((0100 \,\&\, 1100) == (0110 \,\&\, 1100))$$

$$\text{true} \,\&\&\, ((01) == (01))$$

But path '01' is revoked by subset '0'