DNS4EU

DNS4EU for Public anonymization

Robert Sefr | CTO @ Whalebone

Project number: 101095329 21-EU-DIG-EU-DNS
Project name: DNS4EU and European DNS Shield.
This project is co-funded by the European Union.

whalebone

Co-funded by
the European Union

DNS4EU Services

DNS4EU

DNS4EU for Public

- Different kind of protection and filtering
- To be launched in 2025

DNS4EU for Governments

- Hierarchical structure with multi-tenancy managed by the government
- Protecting anything from small municipalities to large hospitals

DNS4EU for Telcos

- Leveraging the DNS4EU technology, intelligence and shared IP
- Can also be offered as added-value service

A journey for the sweet spot between privacy and security

Co-funded by
the European Union

Looking for the sweet spot

Private DNS Resolver

- Don’ log anything!

“No, we would be completely blind.”

Protective DNS Resolver

- Log everything and learn from the logs!

“No, we would see personal information.”

Looking for the sweet spot

Private DNS Resolver

- Don’ log client IP address

“No, we will not understand any query sequences.”

Protective DNS Resolver

- Hash the client IP address

“No, common, hashing the IP is not anywhere anonymous.”

Looking for the sweet spot

Private DNS Resolver

- Keep only the first byte of the IP address

"Too wide dataset with too much traffic to make it useful for research."

Protective DNS Resolver

- Replace IP address with token that can not be brute-forced

"But if anyone identifies the relation IP-token once, it will be identified forever."

Looking for the sweet spot

Private and Protective DNS Resolver

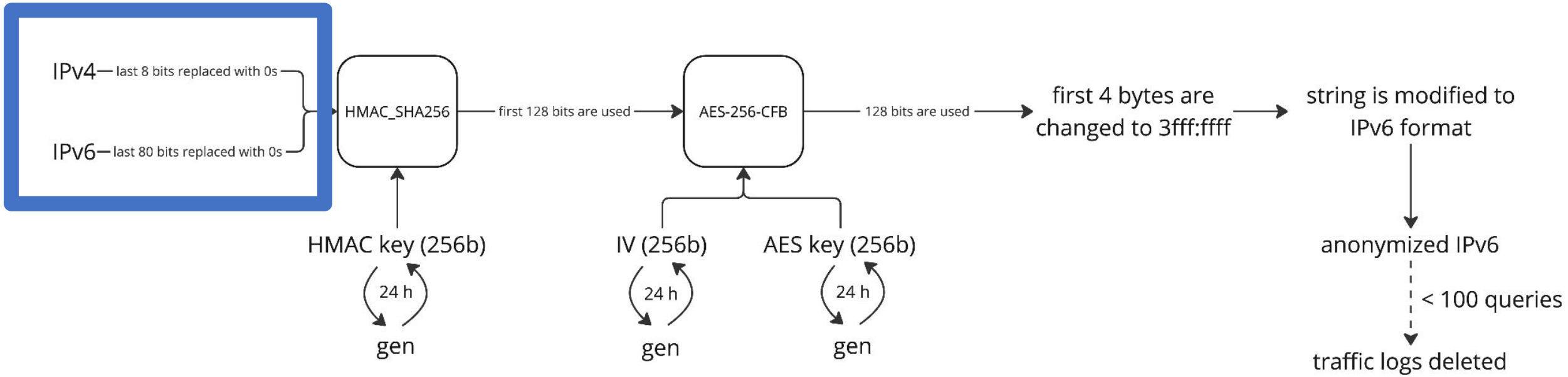
- Generate irreversible token for an IP subnet
- Change the token every 24h
- Drop tiny datasets of particular token

“Well, maybe. But show us the details.”


Anonymization algorithm

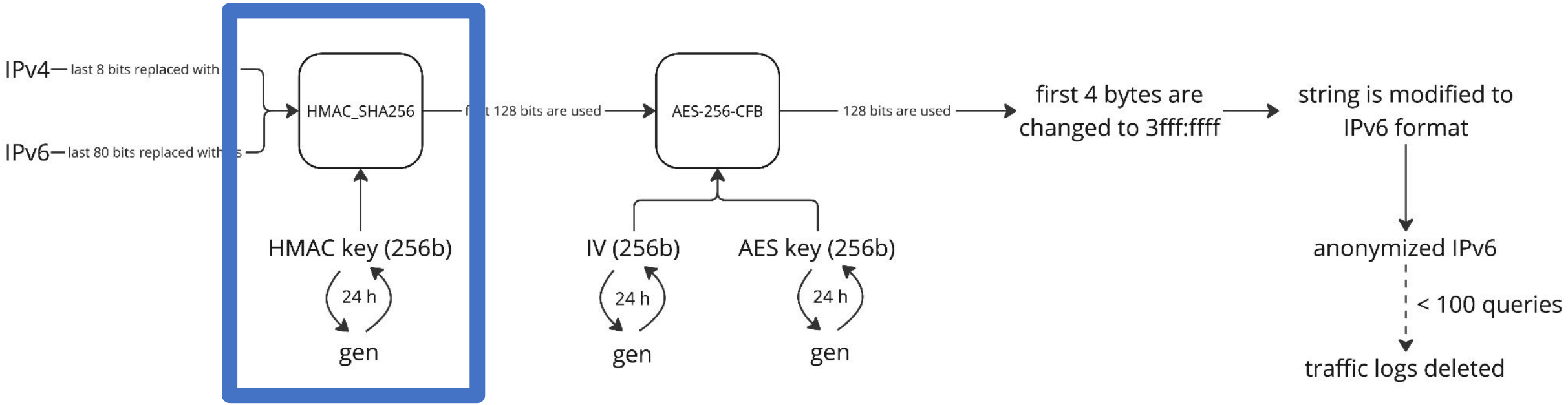
1. Truncate the IP address

- As a very first step drop last 8 bits of IPv4 and/or 80 bits of IPv6 from the client IP



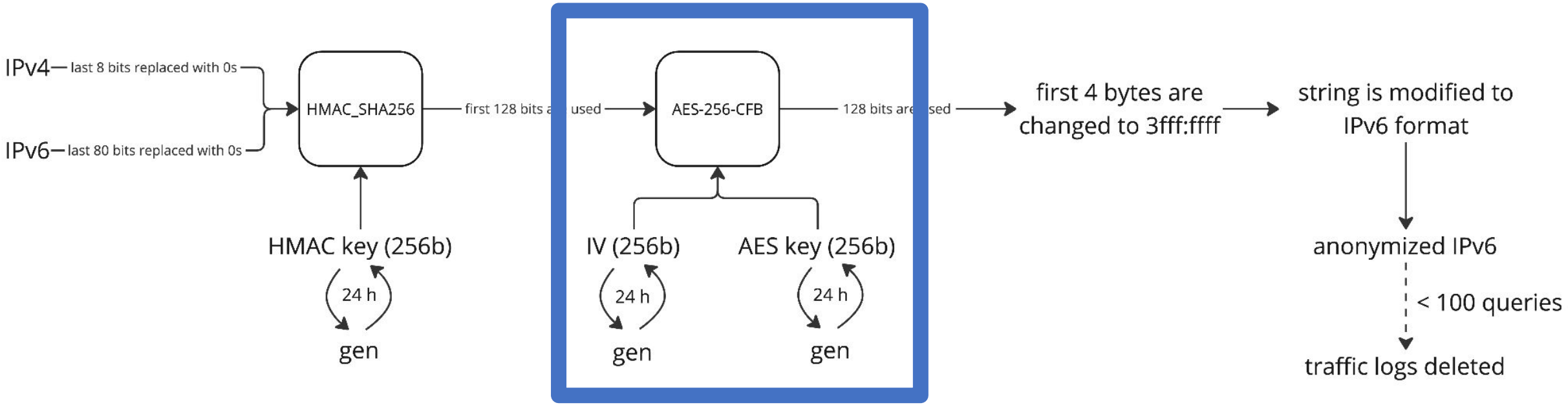
2. Generate a fingerprint

- Truncated IP is hashed (HMAC_SHA256) using a key
- Key is only kept in-memory and regenerated every 24h
- Pass onward only the first 128 bits (half) of the output



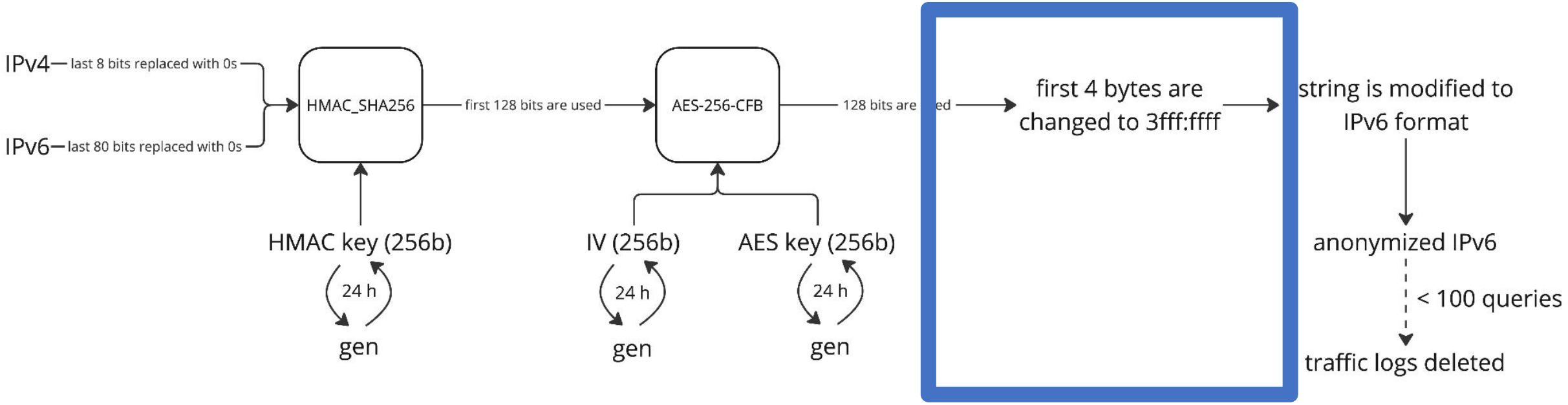
3. Encrypt the truncated fingerprint

- Truncated fingerprint is encrypted (AES-256-CFB)
- Key and IV is only kept in-memory and regenerated every 24h



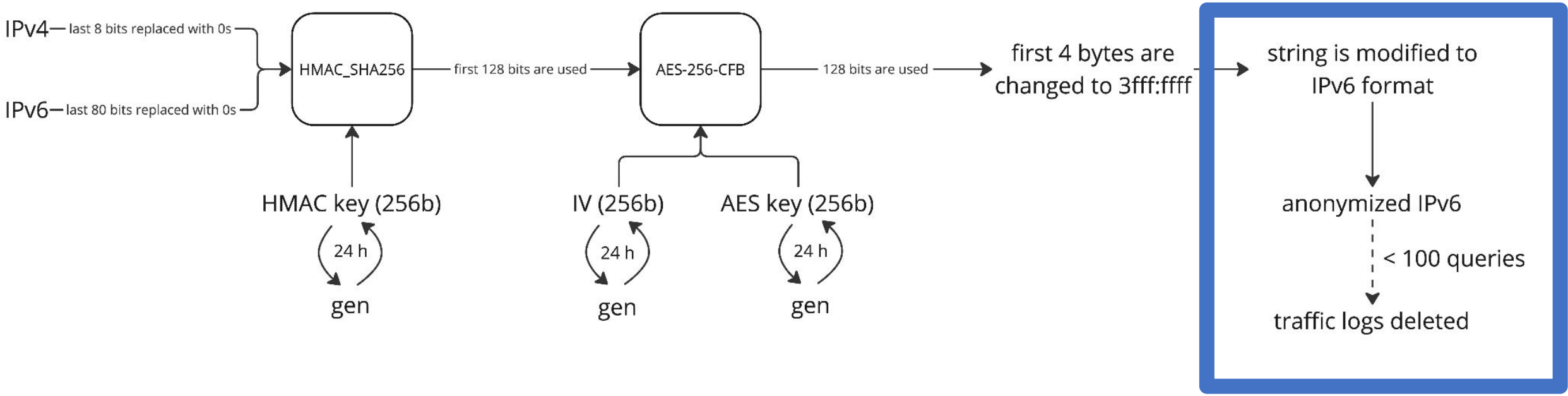
4. Truncate the encrypted fingerprint

- Represent 128 bits output as IPv6
- Replace starting 4 bytes with “3fff:ffff” (documentation subnet)
- This gives us the final anonymized token to be logged

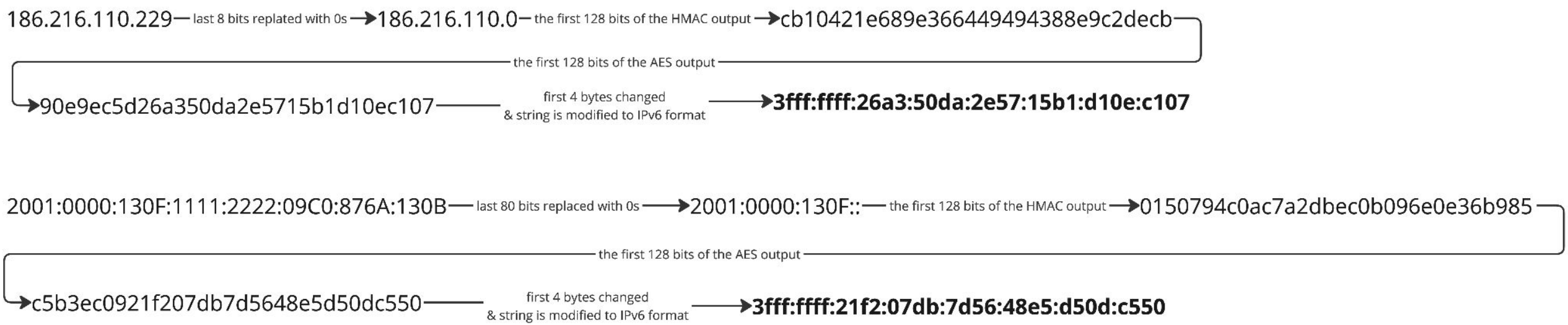


5. Drop small datasets

- At the end of the day, if there are sequences with same token and less than 100 logs, drop them
- Small dataset may include specific identifiable sequences and anonymity has bigger priority than having everything for research



IPv4 and IPv6 step by step example



Final notes

- Keys and IVs are generated per resolver, only kept in memory and not synchronized in any way
- Log of individual clients may end up on different resolvers throughout the day resulting in different fingerprints
- We are very confident that the final fingerprint is irreversible
- In case, both keys and the IV would be compromised
  - The adversary would be able to bruteforce the original subnet, but never a particular IP address
  - Result of such a bruteforce would be limit to single resolver and up to 24h

Questions? Thank you.

Robert Sefr

whalebone.io

robert.sefr@whalebone.io

linkedin.com/in/robertsefr