Mooltipass Mini Signature and Hashing Keys

Introduction:

Several elements are programmed into the Mooltipass Minis during mass production and are unique to each device:
- 2 AES keys (AES key #1 and AES key #2)
- 1 unique identifier request key (UID request key)
- 1 unique identifier (UID)

All these elements are programmed into the microcontroller's eeprom.
All the keys can't be read from the device screen or through USB communications or through debugging interfaces (avrisp, JTAG...). The microcontroller fuses are set in such a way that microcontroller reprogramming is possible, provided a complete device erase is issued, hence deleting all the elements mentioned above.

UID and UID request key

Type	Secret element, password
Size	Request Key: 16 bytes, UID: 6 bytes
Generation Method	True Random Number Generator on an offline PC
Purpose / Protected Element or Feature	Detect flash erasure
Provided Assurance	A valid value ensures the MCU internal flash has not been erased by an external write attempt

These 2 elements are there to make sure the microcontroller hasn't been tampered with between mass production and device reception by the customer.
The UID request key is used to access the UID. Through an authenticated communication channel, the Mooltipass team will provide the customer with both elements so the customer can check that the UID matches and therefore check that the microcontroller memory contents haven't been modified without their previous knowledge.

AES key 1

Type	Crypto Key
Size	256 bits
Generation Method	True Random Number Generator on an offline PC
Purpose / Protected Element or Feature	Ensures upgrade bundle integrity & origin
Provided Assurance	Unit-specific cryptographic CBC-MAC signature of the upgrade bundle by the bootloader's upgrade routine

The role of AES key 1 is to check the signature of the upgrade bundles sent to the Mooltipass device. An upgrade bundle is composed of the following elements:
- updated graphics
- updated firmware
- new firmware version (ASCII)
- AES key 1 update flag
- (valid if previous flag is set) new AES key 1, encrypted
- CBC-MAC signature

An upgrade bundle is of fixed length. The bootloader, in charge of updating the device, will check the bundle signature before updating the device and will also make sure that the firmware version to be flashed is subsequent to the one currently present on the device.
Relevant file: https://github.com/limpkin/mooltipass/blob/master/source_code/src/bootloader_main.c

AES key 2

Type	Crypto Key
Size	256 bits
Generation Method	True Random Number Generator on an offline PC
Purpose / Protected Element or Feature	Unlocks upgrade feature and compute onboard firmware integrity hash
Provided Assurance	A valid value ensures the MCU internal flash has not been erased by an external write attempt

The AES key 2 provides firmware authentication capabilities. Upon user card insertion and unlock, hashes may be displayed to the user. These hashes, unique to each Mooltipass user are computed using this AES key 2.
Relevant function: validCardDetectedFunction(uint16_t* suggested_pin, uint8_t hash_allow_flag) in https://github.com/limpkin/mooltipass/blob/master/source_code/src/LOGIC/logic_smartcard.c
Any tempering with the Mooltipass Mini microcontroller's memory contents will therefore result in different hashes displayed to the user, hence allowing the user to make sure his Mini hasn't been tampered with.
The AES key 2 is also used to generate the password required to send the upgrade bundle to the device. The password is the AESenc(current firmware version | device UID).

Conclusion

In this small document the elements unique to each device are described. As a result, upgrade bundles are generated on a per-device & firmware version basis and the Mooltipass team does not hold a "master key" that would allow a potential attacker to compromise all Mooltipass Minis.
The sequence of events needed to make sure a Mini hasn't been compromised therefore consists in:
- creating a new user on the Mooltipass Mini
- enabling the hash display feature
- remembering the hash #1 and hash #2 displayed on the Mini screen when inserting and unlocking the smart card
- contacting the Mooltipass team to request the UID & UID request key
- using the Mooltipass App and the UID request key to check that the UID is correct
- checking hash #1 and hash #2 when inserting and unlocking the smartcard