

Σχεδιασμός και ανάπτυξη ενός ασφαλούς συστήματος μεταφοράς δεδομένων μέσω BLE σε μικροελεγκτές STM32 με TF-M

Ξενοφών Ραφαήλ Παπαδόπουλος



Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Πολυτεχνική Σχολή
Πανεπιστήμιο Ιωαννίνων

Μάρτιος 2023

ΠΕΡΙΕΧΟΜΕΝΑ

Κατάλογος Σχημάτων	iv
Κατάλογος Πινάκων	vi
Περίληψη	vii
Abstract	ix
1 Εισαγωγή	1
1.1 Στόχοι διπλωματικής	3
1.2 Δομή της διπλωματικής εργασίας	3
2 Θεωρητικό υπόβαθρο	5
2.1 Τι είναι ένας Μικροελεγκτής	5
2.1.1 Ανατομία μικροελεγκτή - Πως δουλεύει	7
2.1.2 Διαφορές Μικροελεγκτή - Μικροεπεξεργαστή	9
2.1.3 Επιλέγοντας τον σωστό μικροελεγκτή	10
2.2 Internet of Things	11
2.2.1 MCUs και IoT	11
2.2.2 Οφέλη	12
2.2.3 Απειλές που προκύπτουν	13
2.3 Ασφάλεια	15
2.3.1 Μοντέλο απειλών	15
2.3.2 Ταξινόμηση επιθέσεων	17
2.3.3 Αντίμετρα	22
2.4 Ασφάλεια με βάση το υλικό	26
2.4.1 Root of Trust	26
2.4.2 Platform Security Architecture (PSA)	26

2.4.3	Trusted execution environment	30
2.4.4	TrustZone απο την ARM	31
2.4.5	Ασφαλής εκκίνηση σε μικροελεγκτές STM32	34
2.5	Trusted Firmware - M	35
2.5.1	X-CUBE-SBSFU vs. TF-M	36
2.5.2	SBSFU με το TF-M	39
2.5.3	Ασφαλείς υπηρεσίες στο TF-M	45
2.5.4	Μέτρα προστασίας και στρατηγική ασφάλειας	48
2.6	Bluetooth Low Energy	52
2.6.1	Bluetooth vs BLE	53
2.6.2	ATT και GATT	54
2.6.3	Advertising data	54
3	Υλοποίηση του Συστήματος	57
3.1	Αρχιτεκτονική συστήματος	57
3.1.1	Το δικό μας μοντέλο απειλών	58
3.1.2	Ο συνδυασμός Λογισμικό-Υλικό που διαλέξαμε	60
3.2	Προγραμματιστικά εργαλεία	61
3.2.1	STM32CubeIDE	61
3.2.2	STM32CubeProgrammer	61
3.2.3	Βιβλιοθήκη STM32Cube_FW_L5	61
3.2.4	Tera Term	62
3.3	Σημαντικά Option Bytes	62
3.4	Βασικές υλοποιήσεις	63
3.4.1	Υλοποίηση Trustzone σε STM32	63
3.4.2	Υλοποίηση TF-M σε STM32	66
3.5	Υλοποιήσεις ασφαλών υπηρεσιών στο TF-M	73
3.5.1	Υλοποίηση Secure Boot	74
3.5.2	Υλοποίηση Secure Firmware Update	75
3.5.3	Υλοποίηση Initial Attestation	77
4	Μετρήσεις	79
4.1	Μετρήσεις χρόνου	79
4.1.1	Χρόνος εκκίνησης	80
4.2	Μετρήσεις χρόνου συγκεκριμένες για το TF-M	82

4.2.1	Χρόνοι Secure Boot	82
4.2.2	Χρόνοι Secure Firmware Update	82
4.2.3	Χρόνοι Initial Attestation	85
4.3	Μετρήσεις κατανάλωσης ενέργειας	87
5	Συμπεράσματα	91
5.1	Συμπεράσματα	91
5.2	Μελλοντικές Επεκτάσεις	92
	Βιβλιογραφία	93
A	Λήψη δεδομένων BLE απο MCU	97
A.1	nRF Connect App	97
A.2	Python App	97
A.2.1	Επιλογή βιβλιοθήκης	98
A.2.2	Υλοποίηση	98
B	X-CUBE-BLE1	101
B.1	Παράδειγμα SampleApp	101
Γ	TrustZone	105
Γ.1	Απενεργοποίηση του TrustZone	105
Γ.2	Αλλαγή Option Bytes runtime	107

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

1.1	Διάγραμμα σειράς λειτουργιών εφαρμογής	2
1.2	Εικονογράφηση συστήματος	2
2.1	Motorola 6801 - Ένας από τους πρώτους μικροελεγκτές	6
2.2	Ανατομία μικροελεγκτή	7
2.3	Παράδειγμα κώδικα SAU	34
2.4	Επισκόπηση TF-M	36
2.5	X-CUBE-SBSFU vs. TF-M	37
2.6	Firmware image keys personalization	39
2.7	Secure Boot χρονοδιάγραμμα εκτέλεσης	40
2.8	Secure Firmware Update χρονοδιάγραμμα εκτέλεσης	42
2.9	Διαδικασία λήψης και εγκατάστασης νέου υλικολογισμικού (Overwrite operation)	44
2.10	Απεικόνιση διαδικασίας υπογραφής εικόνας	45
2.11	Απεικόνιση διαδικασίας κρυπτογράφησης και υπογραφής εικόνας	45
3.1	Αρχική ιδέα της εφαρμογής	57
3.2	Option Bytes DBANK και TZEN	64
3.3	Option Bytes για τις ασφαλής και μη ασφαλής περιοχές	64
3.4	Όλα τα GPIO ρυθμισμένα ως μη ασφαλή εκτος από το LED10	65
3.5	Οι callback συναρτήσεις "HAL_SYSTICK_Callback"	66
3.6	Οι "Error_Handler" συναρτήσεις	66
3.7	Διαδικασία Compile TF-M	68
3.8	Ανίχνευση εισβολής που εμφανίζεται στο Tera Term	70
3.9	TF-M welcome screen display	71
3.10	TFM local loader welcome screen	71

3.11 Διάγραμμα ροής ασφαλούς εκκίνησης και ασφαλούς ενημέρωσης υλικολογισμικού	73
3.12 Διάγραμμα ροής Initial Attestation	77
4.1 STM32L562xx ρυθμίσεις ρολογιού	80
4.2 Secure Boot χρονοδιάγραμμα εκτέλεσης	80
4.3 Σύγκριση χρόνων εκκίνησης	81
4.4 Secure Firmware Update χρονοδιάγραμμα εκτέλεσης	82
4.5 Γραφήματα χρόνων Secure Firmware Update	84
4.6 Γράφημα συνολικών χρόνων εγκατάστασης εικόνων	85
4.7 Γράφημα χρόνων Initial Attestation	86
4.8 Γράφημα αναλυτικών χρόνων προσθήκης των claim	87
4.9 Συσκευή ελέγχου θύρας USB	88
4.10 Γράφημα σύγκρισης έντασης σε κανονική λειτουργία	90
B.1 X-CUBE-BLE1 SampleApp Init	102
B.2 X-CUBE-BLE1 Pinout Configuration	102
B.3 X-CUBE-BLE1 SPI1 Configuration	103
B.4 X-CUBE-BLE1 NVIC Configuration	103
B.5 X-CUBE-BLE1 Mode and Configuration	104
Γ.1 Εκκίνηση από RSS (Τροποποίηση BOOT0)	106

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

2.1	Πίνακας Διαφορές MCU - MPU	10
2.2	Λειτουργίες ασφαλείας PSA Root of Trust	28
2.3	Διαφορές ανάμεσα στα PSA levels	29
2.4	X-CUBE-SBSFU vs. TF-M top-level features	38
3.1	Option Bytes Settings for TF-M	68
4.1	Boot time: plug - unplug cable	81
4.2	Boot time: reset button	81
4.3	Χρόνοι "Προετοιμασίας" και "Επαλήθευσης" ασφαλούς εκκίνησης	82
4.4	Χρόνοι "Προετοιμασίας" και "Επαλήθευσης 1" ασφαλούς ενημέρωσης	83
4.5	Χρόνοι εγκατάστασης (Διαγραφή παλιάς εικόνας) ασφαλούς ενημέρωσης	83
4.6	Χρόνοι εγκατάστασης (Αντιγραφή νέας εικόνας) ασφαλούς ενημέρωσης	83
4.7	Χρόνοι "Επαλήθευσης 2" ασφαλούς ενημέρωσης	84
4.8	Συνολικοί χρόνοι εγκατάστασης secure και non-secure εικόνων	85
4.9	Χρόνοι Initial Attestation	86
4.10	Αναλυτικοί χρόνοι προσθήκης των claim	86
4.11	Αποτελέσματα μέτρησης έντασης TF-M	89
4.12	Αποτελέσματα μέτρησης έντασης TrustZone	90
4.13	Αποτελέσματα μέτρησης χωρίς ασφάλεια	90

ΠΕΡΙΛΗΨΗ

Ξενοφών Ραφαήλ Παπαδόπουλος, Δίπλωμα, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, Μάρτιος 2023.

Σχεδιασμός και ανάπτυξη ενός ασφαλούς συστήματος μεταφοράς δεδομένων μέσω BLE σε μικροελεγκτές STM32 με TF-M.

Επιβλέπων: Βασίλειος Τενέντες, Επίκουρος Καθηγητής.

Η μετάδοση ευαίσθητων δεδομένων μέσω ασύρματων δικτύων έχει αυξηθεί ως αποτέλεσμα της ευρείας υιοθέτησης των συσκευών Internet of Things (IoT). Κατά συνέπεια, υπάρχει επείγουσα ανάγκη για αξιόπιστα και ασφαλή συστήματα μεταφοράς δεδομένων. Αυτή η εργασία προτείνει μια μέθοδο για ασφαλή μετάδοση δεδομένων χρησιμοποιώντας beacons Bluetooth Low Energy (BLE) και έναν μικροελεγκτή STM32 με εγκατεστημένο το TrustZone και το πλαίσιο ασφαλείας TF-M.

Η πλακέτα STM32L562E-DK, η οποία χρησιμοποιεί τον TrustZone ικανό επεξεργαστή Cortex-M33 και έχει πιστοποίηση PSA Level 2 που εξασφαλίζει υψηλό βαθμό ασφάλειας, είναι αυτή που χρησιμοποιείται σε αυτό το σύστημα. Για την προστασία των δεδομένων από μη εξουσιοδοτημένη πρόσβαση ή τροποποίηση, το TF-M προσφέρει ένα ασφαλές περιβάλλον για επεξεργασία και αποθήκευση δεδομένων. Αφού παραλάβει τα δεδομένα από το beacon, η πλακέτα τα μεταδίδει σε έναν φορητό υπολογιστή για μακροπρόθεσμη αποθήκευση και ανάλυση.

Αρχικά, εξετάστηκαν οι ανάγκες ασφαλείας του προτεινόμενου συστήματος πριν αξιολογηθεί η λειτουργικότητα της πλακέτας STM32L562E-DK εντός του πλαισίου ασφαλείας TF-M. Μετά την υλοποίηση του συστήματος, αξιολογήθηκε η απόδοση του συστήματος όσον αφορά την εκκίνηση, την ενημέρωση εικόνων και την κατανάλωση ενέργειας.

Τα πειράματά περιλαμβάνουν χρονισμούς για ασφαλή εκκίνηση TF-M, ασφαλή ενημέρωση υλικολογισμικού TF-M και αρχική επιβεβαίωση TF-M, καθώς και χρόνους εκκίνησης με ενεργοποιημένο και απενεργοποιημένο το TF-M και το Trust-

zone. Επιπλέον, πραγματοποιήθηκαν μετρήσεις της ενέργειας που καταναλώνεται σε διάφορα στάδια της εφαρμογής για να καθοριστεί ποια λειτουργία εξοικονόμησης ενέργειας είναι η πιο αποδοτική.

Τα αποτελέσματα των δοκιμών δείχνουν ότι το προτεινόμενο σύστημα επιτυγχάνει υψηλά επίπεδα ασφάλειας και αξιοπιστίας δεδομένων, διατηρώντας παράλληλα χαμηλή κατανάλωση ενέργειας. Προκειμένου να προστατεύονται τα δεδομένα που μεταδίδονται από το BLE beacon από μη εξουσιοδοτημένη πρόσβαση ή τροποποίηση, το πλαίσιο TF-M προσφέρει ένα ισχυρό περιβάλλον ασφαλείας. Οι δοκιμές αποδεικνύουν επίσης ότι το προτεινόμενο σύστημα προσφέρει αξιόπιστες και γρήγορες ενημερώσεις υλικολογισμικού, αποδεκτά γρήγορους χρόνους εκκίνησης και γρήγορους χρόνους αρχικής πιστοποίησης. Οι μετρήσεις της κατανάλωσης ισχύος του συστήματος σε διάφορα στάδια εφαρμογής δείχνουν επίσης πόσο ενεργειακά αποδοτικό είναι.

Συμπερασματικά, η παρούσα διατριβή προσφέρει μια μέθοδο για ασφαλή μεταφορά δεδομένων για μικροελεγκτές STM32 με χρήση beacons BLE και TF-M. Μια μεγάλη ποικιλία εφαρμογών IoT μπορεί να χρησιμοποιήσει το προτεινόμενο σύστημα, καθώς προσφέρει υψηλό επίπεδο ασφάλειας δεδομένων, αξιοπιστίας και ενεργειακής απόδοσης.

ABSTRACT

Xenofon Rafail Papadopoulos, Diploma, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, March 2023.

Design and development of a secure data transfer system for STM32 microcontrollers using BLE and TF-M.

Advisor: Vasilios Tenentes, Assistant Professor.

The transmission of sensitive data over wireless networks has increased as a result of the widespread adoption of Internet of Things (IoT) devices. Consequently, there is an urgent need for reliable and secure data transfer systems. This work proposes a method for secure data transmission using Bluetooth Low Energy (BLE) beacons and an STM32 microcontroller with TrustZone and the TF-M security framework installed.

The STM32L562E-DK board, which uses the TrustZone capable Cortex-M33 processor and has PSA Level 2 certification that ensures a high degree of security, is the one used in this system. To protect data from unauthorized access or modification, TF-M offers a secure environment for data processing and storage. After receiving the data from the beacon, the board transmits it to a laptop for long-term storage and analysis.

First, the security needs of the proposed system were considered before the functionality of the STM32L562E-DK board within the TF-M security framework was evaluated. After the implementation of the system, the performance of the system in terms of booting, updating images and power consumption was evaluated.

Experiments include timings for TF-M secure boot, TF-M secure firmware update, and TF-M initial confirmation, as well as boot times with TF-M and Trustzone enabled and disabled. In addition, measurements were made of the energy consumed at various stages of the application to determine which energy-saving mode is the most efficient.

Test results show that the proposed system achieves high levels of data security and reliability while maintaining low power consumption. In order to protect the data transmitted by the BLE beacon from unauthorized access or modification, the TF-M framework offers a strong security environment. The tests also demonstrate that the proposed system offers reliable and fast firmware updates, acceptably fast boot times, and fast initial certification times. Measurements of the system's power consumption at various application stages also show how energy efficient it is.

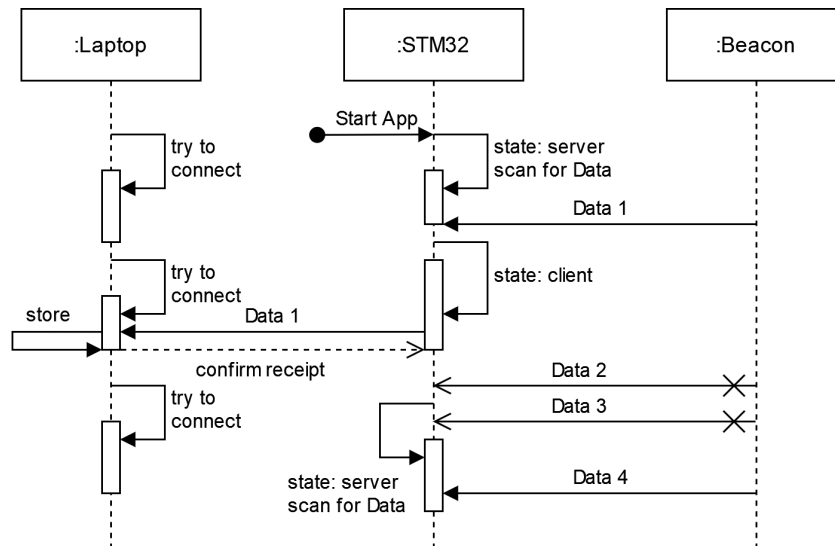
In conclusion, this thesis offers a method for secure data transfer for STM32 microcontrollers using BLE and TF-M beacons. A wide variety of IoT applications can use the proposed system, as it offers a high level of data security, reliability and energy efficiency.

ΚΕΦΑΛΑΙΟ 1

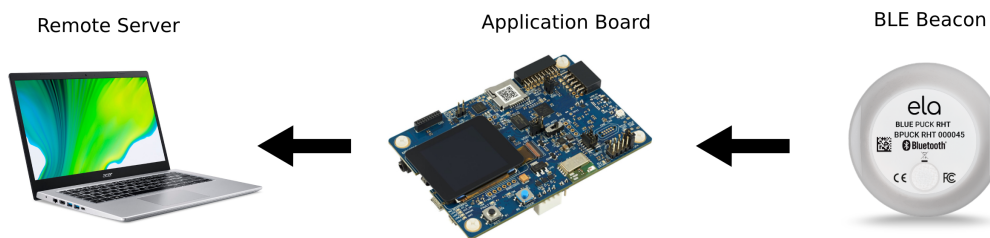
Εισαγωγή

Το Διαδίκτυο των Πραγμάτων (IoT) αλλάζει ραγδαία τον τρόπο που αλληλεπιδρούμε με το περιβάλλον μας. Συνδέοντας συσκευές και αισθητήρες στο διαδίκτυο, συλλέγονται και να αναλύονται δεδομένα σε πραγματικό χρόνο, επιτρέποντάς τη λήψη πιο ενημερωμένων αποφάσεων και τη βελτιστοποίηση διαφόρων διαδικασιών. Ωστόσο, αυτό δημιουργεί επίσης νέες προκλήσεις ασφαλείας που πρέπει να αντιμετωπιστούν για να διασφαλιστεί το απόρρητο και η ακεραιότητα των δεδομένων που μεταδίδονται.

Σε αυτή τη διπλωματική, παρουσιάζετε μια εφαρμογή IoT που χρησιμοποιεί την τεχνολογία Bluetooth Low Energy (BLE) για τη συλλογή δεδομένων από έναν αισθητήρα και τη μετάδοση τους σε έναν απομακρυσμένο διακομιστή. Η εφαρμογή αποτελείται από ένα BLE beacon που στέλνει δεδομένα, μια πλακέτα STM32 που τα συλλέγει και έναν φορητό υπολογιστή που λειτουργεί ως απομακρυσμένος διακομιστής για την αποθήκευση των δεδομένων. Η πλακέτα STM32 λειτουργεί στην αρχή ως διακομιστής και καταγράφει όλα τα δεδομένα που προέρχονται από το beacon. Στη συνέχεια, μεταβαίνει σε λειτουργία πελάτη και δέχεται μια εισερχόμενη σύνδεση από τον απομακρυσμένο διακομιστή, ο οποίος προσπαθεί κάθε δέκα δευτερόλεπτα να συνδεθεί. Αν η σύνδεση πετύχει, το board στέλνει τα δεδομένα στον φορητό υπολογιστή και επιστρέφει στη λειτουργία διακομιστή. Αυτή η διαδικασία επαναλαμβάνεται επ' αόριστον.



Σχήμα 1.1: Διάγραμμα σειράς λειτουργιών εφαρμογής



Σχήμα 1.2: Εικονογράφιση συστήματος

Για τη διασφάλιση της ασφάλειας των δεδομένων που μεταδίδονται, η πλακέτα STM32 έχει εγκαταστημένες τις τεχνολογίες TrustZone και Trusted Firmware-M (TF-M).

Το TrustZone είναι μια λύση υλικού που παρέχει ένα ασφαλές περιβάλλον εκτέλεσης μέσα σε έναν επεξεργαστή. Δημιουργεί έναν ασφαλή κόσμο που είναι απομονωμένος από τον μη ασφαλή κόσμο και αποτρέπει τη μη εξουσιοδοτημένη πρόσβαση σε δεδομένα.

Το TF-M είναι ένα λογισμικό ανοιχτού κώδικα που παρέχει ένα ολοκληρωμένο σύνολο υπηρεσιών ασφαλείας, συμπεριλαμβανομένης της ασφαλούς εκκίνησης, της ασφαλούς ενημέρωσης υλικολογισμικού και της αρχικής επιβεβαίωσης.

Αυτά τα δύο μαζί, παρέχουν μια βάση για την εφαρμογή IoT, προστατεύοντας από πιθανές απειλές, όπως παραβιάσεις δεδομένων, μη εξουσιοδοτημένη πρόσβαση και επιθέσεις κακόβουλου λογισμικού.

1.1 Στόχοι διπλωματικής

- Να προτείνει μια νέα προσέγγιση για ασφαλή μεταφορά δεδομένων χρησιμοποιώντας BLE, το TrustZone και το TF-M σε έναν μικροελεγκτή STM32.
- Να αναλύσει τις απαιτήσεις ασφαλείας του προτεινόμενου συστήματος και να αξιολογήσει την απόδοση του μικροελεγκτή STM32 με TF-M.
- Να υλοποιήσει το προτεινόμενο σύστημα και να αξιολογήσει την αποτελεσματικότητά του όσον αφορά την ασφάλεια των δεδομένων, την αξιοπιστία και την κατανάλωση ενέργειας.
- Η μέτρηση του χρόνου εκκίνησης με το TF-M, χωρίς το TF-M και με το Trustzone απενεργοποιημένο, καθώς και τους χρονισμούς ασφαλούς εκκίνησης TF-M, τους χρονισμούς ενημέρωσης ασφαλούς υλικολογισμικού TF-M και τους χρονισμούς αρχικής επιβεβαίωσης TF-M.
- Η μέτρηση της κατανάλωσης ενέργειας σε διάφορες φάσεις της εφαρμογής, όπως κατά τη διάρκεια μεγάλου φορτίου, κανονικής λειτουργίας και κατάστασης βαθύ ύπνου.
- Να αποδείξει την αποτελεσματικότητα του προτεινόμενου συστήματος στην επίτευξη υψηλών επιπέδων ασφάλειας δεδομένων, αξιοπιστίας και απόδοσης ισχύος.
- Να συμβάλει στην ανάπτυξη ασφαλών και αξιόπιστων συστημάτων μεταφοράς δεδομένων για εφαρμογές IoT.

1.2 Δομή της διπλωματικής εργασίας

Η διπλωματική εργασία περιέχει 5 κεφάλαια, με το καθένα να καλύπτει συγκεκριμένες πτυχές της ασφάλειας των μικροελεγκτών.

Το πρώτο κεφάλαιο είναι μια εισαγωγή που παρέχει μια επισκόπηση της εργασίας και εξηγεί τους στόχους της.

Στο δεύτερο κεφάλαιο, «Θεωρητικό υπόβαθρο», εξηγείτε τι είναι και πώς λειτουργεί ένας μικροελεγκτής, οι διαφορές μεταξύ μικροελεγκτών και μικροεπεξεργαστών και τα κριτήρια επιλογής του σωστού μικροελεγκτή. Αναλύετε επίσης η

έννοια του Διαδικτύου των Πραγμάτων (IoT), τα οφέλη, οι απειλές και οι ανησυχίες που προκύπτουν για την ασφάλεια. Περιγράφετε περαιτέρω διαφορετικά μοντέλα απειλών, την ταξινόμηση απειλών σε μικροελεγκτές και αντίμετρα που μπορούν να υλοποιηθούν, συμπεριλαμβανομένων μέτρων που βασίζονται σε υλικό. Εξηγεί το Root of Trust, το Platform Security Architecture (PSA), τα Trusted execution environments και περιγράφει τις λύσεις TrustZone από την ARM και TF-M. Το κεφάλαιο ολοκληρώνεται με μια αναφορά στην τεχνολογία Bluetooth Low Energy (BLE).

Στο τρίτο κεφάλαιο, «Υλοποίηση του Συστήματος», παρουσιάζετε η αρχιτεκτονική του συστήματος και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση, συμπεριλαμβανομένου του μοντέλου απειλής, του συνδυασμού υλικού-λογισμικού που επιλέχθηκε και των εργαλείων προγραμματισμού. Περιλαμβάνετε επίσης μια υποενότητα για την περιγραφή της υλοποίησης ασφαλών υπηρεσιών όπως η ασφαλής εκκίνηση, η ασφαλής ενημέρωση υλικολογισμικού και η αρχική επιβεβαίωση.

Το τέταρτο κεφάλαιο, «Μετρήσεις», καλύπτει τα πειραματικά αποτελέσματα που προέκυψαν από το σύστημα. Αναφέρει τις μετρήσεις κατανάλωσης χρόνου και ενέργειας και συγκρίνει τους χρόνους ασφαλούς εκκίνησης, ασφαλούς ενημέρωσης υλικολογισμικού και Initial Attestation που επιτεύχθηκαν στην υλοποίηση TF-M.

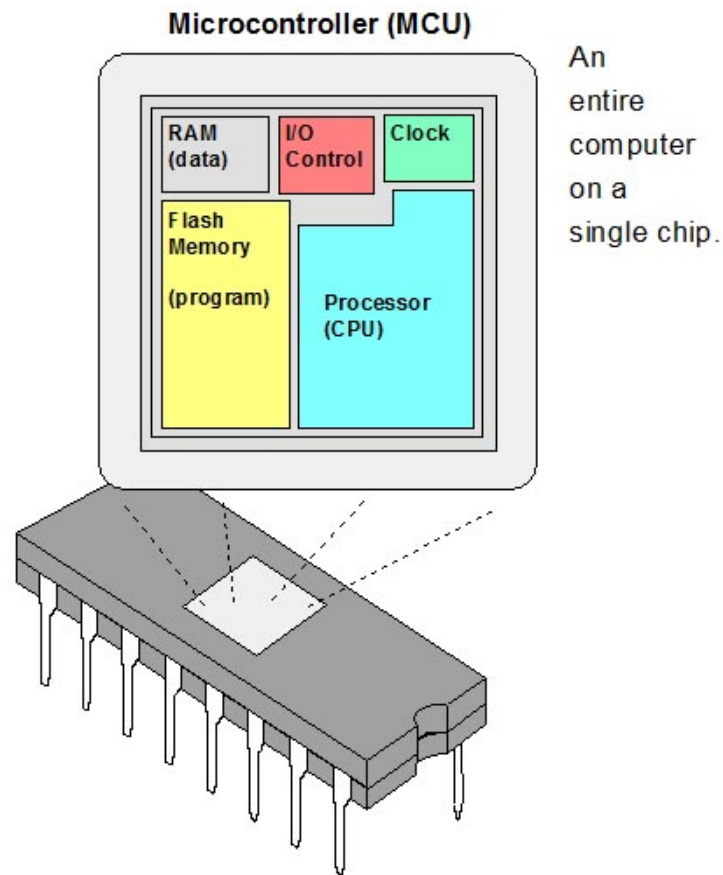
ΚΕΦΑΛΑΙΟ 2

Θεωρητικό υπόβαθρο

2.1 Τι είναι ένας Μικροελεγκτής

Ένας μικροελεγκτής, ή Micro-Controller Unit (MCU) στα αγγλικά, είναι ένας μικρός υπολογιστής σε ένα τσιπ ενιαίου ολοκληρωμένου κυκλώματος (IC) αλλά οι πόροι του και οι δυνατότητες του (πχ. μνήμη, ταχύτητα) είναι περιορισμένα αφού προορίζεται για ενσωματωμένες εφαρμογές. Μπορεί να περιέχει ένα ή περισσότερους CPU μαζί με μνήμη προσωρινής αλλά και μόνιμης αποθήκευσης και προγραμματιζόμενα περιφερειακά εισόδου/εξόδου. Στη σύγχρονη ορολογία, ένας μικροελεγκτής είναι παρόμοιος, αλλά λιγότερο εξελιγμένος από ένα σύστημα σε ένα τσιπ (SoC).

Ουσιαστικά, ένας μικροελεγκτής συλλέγει δεδομένα εισόδου, επεξεργάζεται αυτές τις πληροφορίες και πράτει μια συγκεκριμένη ενέργεια με βάση τις πληροφορίες που σύλλεξε. Οι μικροελεγκτές λειτουργούν συνήθως σε χαμηλότερες ταχύτητες, γύρω στο εύρος 1 MHz έως 200 MHz, και πρέπει να σχεδιαστούν ώστε να καταναλώνουν λιγότερη ενέργεια, επειδή είναι ενσωματωμένοι σε άλλες συσκευές που μπορούν να έχουν μεγαλύτερη κατανάλωση ενέργειας σε άλλες περιοχές.

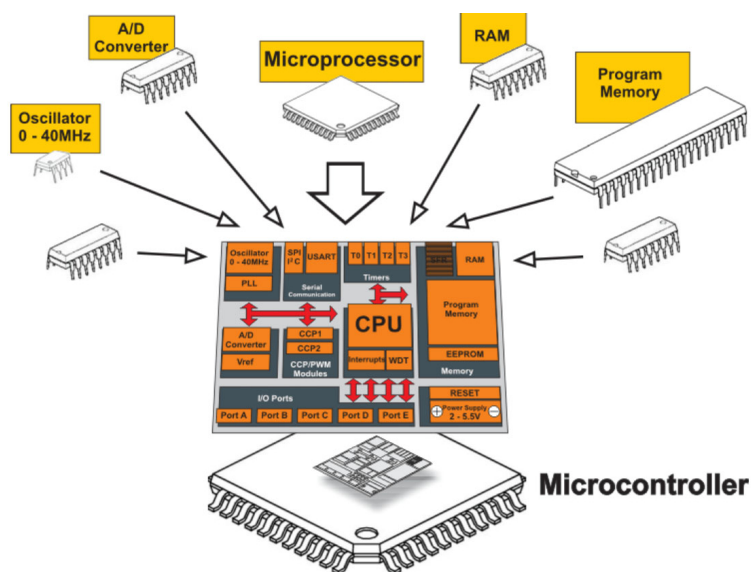


Σχήμα 2.1: Motorola 6801 - Ένας από τους πρώτους μικροελεγκτές

Οι μικροελεγκτές χρησιμοποιούνται σε αυτόματα ελεγχόμενα προϊόντα και συσκευές, όπως συστήματα ελέγχου κινητήρα αυτοκινήτων, ιατρικές συσκευές, τηλεχειριστήρια, μηχανές γραφείου, ηλεκτρικά εργαλεία, παιχνίδια και άλλα ενσωματωμένα συστήματα. Μειώνοντας το μέγεθος και το κόστος σε σύγκριση με ένα σχέδιο που χρησιμοποιεί ξεχωριστό μικροεπεξεργαστή, μνήμη και συσκευές εισόδου/εξόδου, οι μικροελεγκτές καθιστούν οικονομικό τον ψηφιακό έλεγχο ακόμη περισσότερων συσκευών και διεργασιών. Οι μικροελεγκτές μικτού σήματος είναι συνηθισμένοι, οι οποίοι ενσωματώνουν αναλογικά στοιχεία που απαιτούνται για τον έλεγχο μην ψηφιακών ηλεκτρονικών συστημάτων. Στο πλαίσιο του Διαδικτύου των πραγμάτων (IoT), οι μικροελεγκτές είναι ένα οικονομικό και δημοφιλές μέσο συλλογής δεδομένων, ανίχνευσης και ενεργοποίησης του φυσικού κόσμου ως συσκευές αιχμής.

2.1.1 Ανατομία μικροελεγκτή - Πως δουλεύει

Τα βασικά εξαρτήματα στο εσωτερικό του ενός μικροελεγκτή είναι η Κεντρική Μονάδα Επεξεργασίας (CPU), η Μνήμη Τυχαίας Προσπέλασης (RAM), η Μνήμη Flash, η Διασύνδεση Σειριακού Διαύλου, οι Θύρες Εισόδου/Εξόδου (Θύρες I/O) και σε πολλές περιπτώσεις, η Ηλεκτρικά Διαγραφόμενη Προγραμματιζόμενη Μνήμη Μόνο για Ανάγνωση (EEPROM). Τα κύρια εξαρτήματα αλλά και άλλα εξαρτήματα του μικροελεγκτή απεικονίζονται με μεγάλη λεπτομέρεια στο Σχήμα 2.2. Ας εξετάσουμε κάθε ένα από αυτά τα στοιχεία με περισσότερες λεπτομέρειες και ας μάθουμε πώς τα χρησιμοποιεί ο μικροελεγκτής.[1]



Σχήμα 2.2: Ανατομία μικροελεγκτή

Η CPU, που μερικές φορές ονομάζεται επεξεργαστής ή μικροεπεξεργαστής, ελέγχει όλες τις εντολές που λαμβάνει ο μικροελεγκτής. Θεωρείτε ο εγκέφαλος του συστήματος, που επεξεργάζεται όλα τα δεδομένα και εκτελεί τις απαιτούμενες οδηγίες. Τα δύο κύρια στοιχεία του είναι η Αριθμητική Λογική Μονάδα (ALU), η οποία εκτελεί αριθμητικές και λογικές πράξεις και η Μονάδα Ελέγχου (CU), η οποία χειρίζεται όλες τις εκτελέσεις εντολών του επεξεργαστή.

Η μνήμη RAM αποθηκεύει προσωρινά δεδομένα και μπορεί να προσπελαστεί γρήγορα. Παρέχει γρήγορη πρόσβαση ανάγνωσης και εγγραφής στη συσκευή αποθήκευσης. Αυτό διαφέρει από τις περισσότερες άλλες μνήμες, καθώς χρειάζονται περισσότερο χρόνο για την εξαγωγή δεδομένων, αφού τα δεδομένα δεν είναι άμεσα διαθέσιμα. Η μνήμη RAM βελτιώνει τη συνολική απόδοση του συστήματος επειδή

επιτρέπει στον μικροελεγκτή να λειτουργεί ταυτόχρονα με περισσότερες πληροφορίες. Το περιεχόμενό της διαγράφεται πάντα όταν απενεργοποιείται ο μικροελεγκτής.

Η μνήμη Flash είναι ένας τύπος μνήμης που, σε αντίθεση με τη μνήμη RAM, διατηρεί τα δεδομένα της για μεγάλο χρονικό διάστημα, ακόμη και αν ο μικροελεγκτής είναι απενεργοποιημένος, και για αυτό εκεί είναι αποθηκευμένο το πρόγραμμα που εκτελεί ο μικροελεγκτής. Η μνήμη Flash είναι δομημένη σε blocks και όταν γράφει ο χρήστης σε αυτή, γράφει σε ένα μπλοκ κάθε φορά. Αν χρειαστεί να ξαναγραφτεί μόνο ένα byte, η μνήμη Flash θα χρειαστεί να ξαναγράψει ολόκληρο το μπλοκ στο οποίο βρίσκεται το byte.[2]

Η EEPROM είναι σαν τη μνήμη Flash, καθώς έχει την ικανότητα να διατηρεί τα δεδομένα της ακόμα και μετά τον τερματισμό της λειτουργίας του μικροελεγκτή. Η διαφορά είναι ότι, ενώ η Flash ξαναγράφει ένα ολόκληρο μπλοκ για να ξαναγράψει ένα byte, η EEPROM μπορεί να ξαναγράψει οποιοδήποτε συγκεκριμένο byte ανά πάσα στιγμή. Αυτό επεκτείνει τη διάρκεια ζωής της EEPROM σε σύγκριση με τη μνήμη Flash, αλλά σημαίνει επίσης ότι είναι πιο ακριβή.

Η διασύνδεση σειριακού διαύλου (Serial Bus Interface) είναι η σειριακή επικοινωνία στον μικροελεγκτή, που στέλνει δεδομένα ένα bit τη φορά. Συνδέει όλα τα ολοκληρωμένα κυκλώματα σε μια πλακέτα τυπωμένου κυκλώματος (PCB) έτσι μεταφέρουν δεδομένα μεταξύ τους και μειώνουν τον αριθμό των pin σε ένα πακέτο καθιστώντας τα πιο οικονομικά. Παραδείγματα σειριακών διαύλων σε ολοκληρωμένα κυκλώματα είναι το SPI και το I2C.

Οι θύρες I/O είναι αυτές που χρησιμοποιεί ο μικροελεγκτής για να συνδεθεί με εξωτερικές συσκευές. Οι είσοδοι μπορεί να είναι, η αλλαγή θερμοκρασίας από έναν αισθητήρα θερμότητας, η ανίχνευση κίνησης, το πάτημα ενός κουμπιού και πολλά άλλα. Η είσοδος πηγαίνει στη συνέχεια στη CPU και αυτή αποφασίζει τι να κάνει με όλες αυτές τις πληροφορίες. Όταν έρθει η ώρα να εκτελεστεί μια συγκεκριμένη εντολή με βάση μια συγκεκριμένη τιμή από την είσοδο, η CPU στέλνει ένα σήμα στις θύρες εξόδου, όπου αυτό μπορεί να είναι σβήσε μια λυχνία LED, γύρνα έναν κινητήρα έως ένα συγκεκριμένο σημείο, αλλά και πολλά άλλα.

Οι μικροελεγκτές έχουν σχεδιαστεί για ενσωματωμένες εφαρμογές, σε αντίθεση με τους μικροεπεξεργαστές που χρησιμοποιούνται σε προσωπικούς υπολογιστές ή άλλες εφαρμογές γενικής χρήσης που αποτελούνται από διάφορα διακριτά τσιπ.

2.1.2 Διαφορές Μικροελεγκτή - Μικροεπεξεργαστή

Ένας μικροελεγκτής (MCU) είναι ένας μικρός υπολογιστής σε ένα ενιαίο ολοκληρωμένο κύκλωμα που περιέχει έναν πυρήνα επεξεργαστή, μνήμη και προγραμματιζόμενα περιφερειακά εισόδου/εξόδου. Ένας μικροεπεξεργαστής (MPU), από την άλλη, ενσωματώνει τις λειτουργίες της κεντρικής μονάδας επεξεργασίας ενός υπολογιστή (CPU) σε ένα ενιαίο ολοκληρωμένο κύκλωμα (IC). Με απλά λόγια, ο μικροελεγκτής είναι ένας πλήρης υπολογιστής σε ένα μόνο τσιπ ενώ ο μικροεπεξεργαστής είναι η CPU του υπολογιστή σε ένα μόνο τσιπ. Στον πίνακα 2.1 φαίνονται αναλυτικότερα η διαφορές μεταξύ Μικροεπεξεργαστή και Μικροελεγκτή.[3]

Μικροελεγκτής	Μικροεπεξεργαστής
Ο μικροελεγκτής είναι η καρδιά ενός ενσωματωμένου συστήματος.	Ο μικροεπεξεργαστής είναι η καρδιά του υπολογιστικού συστήματος.
Η μνήμη και το I/O υπάρχουν ήδη και το εσωτερικό κύκλωμα είναι μικρό.	Η μνήμη και το I/O πρέπει να συνδεθούν εξωτερικά, έτσι το κύκλωμα γίνεται μεγάλο.
Διαθέτει CPU μαζί με RAM, ROM και άλλα περιφερειακά ενσωματωμένα σε ένα μόνο τσιπ.	Δεν έχει RAM, ROM, ρολόι και άλλα περιφερειακά στο τσιπ.
Το σύστημα είναι απλό με λιγότερο αριθμό εντολών για επεξεργασία.	Το σύστημα είναι περίπλοκο με μεγάλο αριθμό οδηγιών για επεξεργασία.
Το κόστος ολόκληρου του συστήματος είναι χαμηλό.	Το κόστος ολόκληρου του συστήματος είναι υψηλό.
Οι περισσότεροι προσφέρουν λειτουργία εξοικονόμησης ενέργειας.	Οι περισσότεροι δε διαθέτουν δυνατότητες εξοικονόμησης ενέργειας.
Έχει περισσότερους καταχωρητές οπότε τα προγράμματα είναι πιο εύκολο να γραφτούν.	Έχει μικρότερο αριθμό καταχωρητών επομένως περισσότερες λειτουργίες βασίζονται στη μνήμη.
Τα συστήματα λειτουργούν έως και 200 MHz ή περισσότερα ανάλογα με την αρχιτεκτονική.	Τα συστήματα μπορούν να λειτουργούν με πολύ υψηλή ταχύτητα λόγω της τεχνολογίας που εμπλέκεται.
Χρησιμοποιείται κυρίως σε ενσωματωμένα συστήματα.	Χρησιμοποιείται κυρίως σε προσωπικούς υπολογιστές.

Χρησιμοποιείται για ειδικές εφαρμογές.	Χρησιμοποιείται για εφαρμογές γενικού σκοπού που επιτρέπουν πολλά δεδομένα.
--	---

Πίνακας 2.1: Πίνακας Διαφορές MCU - MPU

Επειδή οι μικροελεγκτές είναι κατάλληλοι για συγκεκριμένες εργασίες, είναι απαραίτητο να επιλεγεί έναν μικροελεγκτή που είναι πιο κατάλληλος για το πρότζεκτ. Υπάρχουν πολλοί παράγοντες που πρέπει να ληφθούν υπόψη.[4]

2.1.3 Επιλέγοντας τον σωστό μικροελεγκτή

- **Απόδοση ισχύος:** Υπάρχει μια αντιστάθμιση μεταξύ της απόδοσης επεξεργασίας και της κατανάλωσης ενέργειας, μια συσκευή με υψηλότερη ισχύ επεξεργασίας θα καταναλώνει περισσότερη ενέργεια. Επομένως, εάν ο μικροελεγκτής είναι ασύρματος και λειτουργεί με επαναφορτιζόμενη μπαταρία, είναι απαραίτητο να θυσιαστεί η απόδοση ισχύος έναντι της πλεονάζουσας ισχύος επεξεργασίας ή το αντίστροφο.
- **Ανοχή θερμοκρασίας:** Ανάλογα με το περιβάλλον στο οποίο λειτουργεί ο μικροελεγκτής, μπορεί να χρειάζεται αντοχή σε ακραίες θερμοκρασίες.
- **Μνήμη:** Η ποσότητα μνήμης (RAM και Flash) που χρειάζεται εξαρτάται από τα προγράμματα που εκτελεί. Περισσότερες εργασίες και δεδομένα απαιτούν περισσότερη μνήμη τυχαίας πρόσβασης (RAM) και αποθηκευτικό χώρο.
- **Κόστος:** Οι μικροελεγκτές εμπίπτουν σε ένα ευρύ φάσμα τιμών με τα χαρακτηριστικά που προσφέρουν, από εκατό μονάδες για λίγα ευρώ έως μερικά ευρώ ανά μονάδα ανάλογα. Για πρότζεκτ που απαιτούν πολλές συσκευές πρέπει να ληφθεί υπόψιν το κόστος.
- **Ασφάλεια:** Η πειρατεία που στοχεύει συσκευές IoT αυξάνεται. Σε απάντηση, οι κατασκευαστές μικροελεγκτών εφαρμόζουν επίπεδα ασφάλειας όπως η κρυπτογραφία και η φυσική ασφάλεια που θα αναπτυχθούν στη συνέχεια.

2.2 Internet of Things

Το Διαδίκτυο των Πραγμάτων, ή IoT, είναι ένα σύστημα αλληλένδετων υπολογιστικών συσκευών, μηχανικών και ψηφιακών μηχανών, αντικειμένων, ζώων ή ανθρώπων που παρέχονται με μοναδικά αναγνωριστικά (UID) και τη δυνατότητα μεταφοράς δεδομένων μέσω ενός δικτύου χωρίς να απαιτείται αλληλεπίδραση ανθρώπου με ανθρώπου ή ανθρώπου με υπολογιστή.[5]

Η νέα αυτή κατηγορία υπηρεσιών αυξάνει την εξάρτησή μας από την τεχνολογία του διαδικτύου σε πολύ μεγαλύτερο βαθμό από ότι θα ήταν δυνατή μόλις πριν από δέκα χρόνια. Αυτό οφείλεται στο γεγονός ότι τόσο οι υπηρεσίες cloud όσο και οι υπηρεσίες πληροφοριών εσωτερικής εγκατάστασης έχουν γίνει ουσιαστικά στοιχεία της σύγχρονης ζωής. Οι υπηρεσίες IoT επιτρέπουν την επικοινωνία μεταξύ κοινών gadget όπως οικιακές συσκευές, ηλεκτρονικά είδη ευρείας κατανάλωσης, βιομηχανικά χειριστήρια και αισθητήρες.

Αυτά τα συστήματα και οι υπηρεσίες IoT προσφέρουν υψηλότερο επίπεδο αυτοματισμού και λειτουργικότητας από ότι ήταν προηγουμένως εφικτό μέσω της αλληλεπίδρασης με πιο συμβατικές συσκευές συνδεδεμένες στο Διαδίκτυο, όπως διακομιστές και δρομολογητές.

Σύμφωνα με διάφορες πηγές, το μέγεθος της παγκόσμιας αγοράς IoT αναμένεται να αυξηθεί σημαντικά τα επόμενα χρόνια, με ετήσιο ρυθμό ανάπτυξης να κυμαίνεται από 10,53% έως 26,9%. Το μέγεθος της αγοράς αποτιμήθηκε σε περίπου 330-385 δισεκατομμύρια δολάρια το 2020 και προβλέπεται να φτάσει περίπου τα 875-2465 δισεκατομμύρια δολάρια έως το 2025 και περίπου τα 1600-2500 δισεκατομμύρια δολάρια έως το 2029.[6][7][8]

2.2.1 MCUs και IoT

Οι περισσότεροι αισθητήρες από μόνοι τους δεν προσφέρουν χρήσιμη πληροφορία ενώ συνδέοντας τον κατάλληλο μικροελεγκτή δύναται η πλήρης αξιοποίηση των δυνατοτήτων τους. Πιο συγκεκριμένα, ένας αισθητήρας κίνησης και ένα μοτεράκι ξεχωριστά δεν είναι λειτουργικά, με τη σύνδεση αυτών σε ένα arduino μπορεί να φτιάξει κανείς μια αυτόματη κλειδαριά.

Στη γεωργία, οι αισθητήρες IoT μπορούν να χρησιμοποιηθούν για την παρακολούθηση των επιπέδων υγρασίας του εδάφους, της θερμοκρασίας και άλλων περιβαλλοντικών παραγόντων, για τη βελτιστοποίηση της ανάπτυξης και της απόδοσης

των καλλιεργειών. Στην υγειονομική περίθαλψη, οι συσκευές IoT μπορούν να παρακολουθούν τα ζωτικά σημεία των ασθενών εξ αποστάσεως, επιτρέποντας πιο εξατομικευμένη φροντίδα και ταχύτερους χρόνους απόκρισης έκτακτης ανάγκης. Στις μεταφορές, η τεχνολογία IoT μπορεί να χρησιμοποιηθεί για την παρακολούθηση οχημάτων, τη βελτιστοποίηση διαδρομών και τη βελτίωση της απόδοσης καυσίμου. Τα έξυπνα σπίτια είναι ένας άλλος τομέας όπου το IoT αφήνει το σημάδι του, με συνδεδεμένες συσκευές που επιτρέπουν τον απομακρυσμένο έλεγχο των οικιακών συσκευών, της θέρμανσης και του φωτισμού. Αυτά είναι μόνο μερικά παραδείγματα από τους πολλούς τρόπους με τους οποίους το IoT μεταμορφώνει τον κόσμο μας, καθιστώντας τον πιο συνδεδεμένο, αποτελεσματικό και έξυπνο.

Ήδη παρατηρείται ότι τα προτερήματα που προσφέρει η σύνδεση τέτοιων συσκευών, στη σύγχρονη εποχή, είναι πολλά.

2.2.2 Οφέλη

Το IoT αναφέρεται σε όλους τους διασυνδεδεμένους αισθητήρες, όργανα και άλλες συσκευές, που σε συνδυασμό με βιομηχανικές εφαρμογές, συμπεριλαμβανομένης της παραγωγής και διαχείρισης ενέργειας, δημιουργούν ένα σύνθετο δίκτυο υπηρεσιών, το οποίο επιτρέπει τον αυτοματισμό σε υψηλότερο επίπεδο.[9] Αυτή η συνδεσιμότητα επιτρέπει τη συλλογή, ανταλλαγή και ανάλυση δεδομένων, καθώς βελτιώνει την απόδοση σε όλη την αλυσίδα παραγωγής. Επιτρέπει επίσης στον τομέα που χρησιμοποιείται να κάνει τεράστια καινοτόμα άλματα, να αποκτήσει σημαντική εξωστρέφεια και να αναπτύξει δραστηριότητες, που πριν ήταν αδύνατον. Τα ακόλουθα είναι μερικά από τα βασικά οφέλη και πλεονεκτήματα του IoT:

Αυξημένη αποτελεσματικότητα και παραγωγικότητα: Οι συσκευές και τα συστήματα IoT έχουν σχεδιαστεί για να αυτοματοποιούν εργασίες, μειώνοντας την ανάγκη για χειροκίνητη παρέμβαση και ελευθερώνοντας χρόνο για πιο σημαντικές εργασίες. Αυτές οι συσκευές μπορούν επίσης να προγραμματιστούν για να εκτελούν συγκεκριμένες εργασίες ή ρουτίνες, αυξάνοντας περαιτέρω την απόδοση και την παραγωγικότητα.

Καλύτερη λήψη αποφάσεων: Οι συσκευές IoT μπορούν να συλλέγουν και να αναλύουν μεγάλους όγκους δεδομένων, παρέχοντας σε επιχειρήσεις και οργανισμούς

πολύτιμες πληροφορίες. Η ικανότητα συλλογής και ανάλυσης δεδομένων σε πραγματικό χρόνο επιτρέπει επίσης καλύτερες αποφάσεις που βασίζονται σε δεδομένα.

Εξοικονόμηση κόστους: Οι συσκευές IoT μπορούν να βοηθήσουν οργανισμούς και ιδιώτες να εξοικονομήσουν χρήματα μειώνοντας τη σπατάλη, βελτιώνοντας την αποτελεσματικότητα και μειώνοντας την ανάγκη για χειροκίνητη παρέμβαση.

Βελτίωση στην ασφάλεια: Οι συσκευές IoT μπορούν να χρησιμοποιηθούν για την παρακολούθηση και την προστασία κρίσιμων υποδομών, όπως δίκτυα ηλεκτρικής ενέργειας, συστήματα μεταφοράς και παροχές νερού. Οι συσκευές IoT μπορούν επίσης να χρησιμοποιηθούν για τη βελτίωση της ασφάλειας των βιομηχανικών λειτουργιών εντοπίζοντας και αποτρέποντας επικίνδυνες καταστάσεις.

Παρά τα προφανή οφέλη που ισχυρίζονται ότι προσφέρουν οι συσκευές και οι υπηρεσίες IoT, θα πρέπει να είναι σαφές ότι οι ανησυχίες για την ασφάλεια που σχετίζονται με την πρόσβαση στο Διαδίκτυο προκαλούν ανησυχία. Είναι γνωστό ότι υπάρχουν πολυάριθμοι κίνδυνοι στο Διαδίκτυο και συνδέοντας κοινά gadget, αυτές οι απειλές μπορούν να φτάσουν ένα ευρύτερο κοινό.[10]

2.2.3 Απειλές που προκύπτουν

Ο αυξανόμενος αριθμός συσκευών IoT φέρνει επίσης νέους κινδύνους για την ασφάλεια. Θα τις αναλύσουμε παρακάτω πιο αναλυτικά αλλά μερικές από τις πιο κοινές απειλές που σχετίζονται με συσκευές IoT περιλαμβάνουν:

- Έλλειψη ενημερώσεων λογισμικού: Οι συσκευές IoT συχνά δεν έχουν τη δυνατότητα να λαμβάνουν ενημερώσεις λογισμικού, γεγονός που τις αφήνει ευάλωτες σε εκμεταλλεύσεις ακόμη και μετά την ανακάλυψη των τρωτών σημείων.
- Malware και viruses: Το κακόβουλο λογισμικό μπορεί να μολύνει συσκευές IoT και να εξαπλωθεί σε άλλες συσκευές του δικτύου.
- Man-in-the-middle επιθέσεις: Οι εισβολείς μπορούν να υποκλέψουν επικοινωνίες μεταξύ συσκευών IoT και να κλέψουν ευαίσθητες πληροφορίες.
- Παραβιάσεις απορρήτου: Οι συσκευές IoT συλλέγουν και αποθηκεύουν μεγάλο όγκο προσωπικών δεδομένων, τα οποία μπορούν να χρησιμοποιηθούν για κακόβουλους σκοπούς, εάν πέσουν σε λάθος χέρια.

- Φυσική κλοπή: Οι συσκευές IoT μπορούν να κλαπούν, δίνοντας στους εισβολείς πρόσβαση σε ευαίσθητες πληροφορίες που είναι αποθηκευμένες στη συσκευή.

Είναι σημαντικό να ληφθούν τα απαραίτητα μέτρα για την ασφάλεια των συσκευών IoT, όπως η χρήση ισχυρών κωδικών πρόσβασης, η τακτική ενημέρωση λογισμικού και η επαγρύπνηση σχετικά με τους τύπους δεδομένων που συλλέγουν και αποθηκεύουν οι συσκευές IoT.

Η τεχνολογία TrustZone από την ARM και το Trusted Firmware-M (TF-M) στοχεύουν να αντιμετωπίσουν το ζήτημα αυτά και να παρέχουν λύσεις για τη διασφάλιση της ακεραιότητας και του απορρήτου των ευαίσθητων πληροφοριών και στοιχείων σε συνδεδεμένες συσκευές και συστήματα. Αυτές οι τεχνολογίες μπορούν να συνεργαστούν για να παρέχουν μια ασφαλή βάση για την ανάπτυξη ασφαλών εφαρμογών και υπηρεσιών.

Διάσημα Παραδείγματα Hack

Jeep Cherokee Hack: Το 2015, οι ερευνητές ασφαλείας Charlie Miller και Chris Valasek απέδειξαν ότι μπορούσαν να ελέγξουν εξ αποστάσεως ένα Jeep Cherokee εκμεταλλευόμενοι τα τρωτά σημεία στο σύστημα ψυχαγωγίας του οχήματος. Οι ερευνητές κατάφεραν να αποκτήσουν τον έλεγχο του συστήματος ψυχαγωγίας του οχήματος, το οποίο ήταν συνδεδεμένο στο Διαδίκτυο, και να χειριστούν τις λειτουργίες του, όπως τον κλιματισμό, το ραδιόφωνο, ακόμη και την επιτάχυνση του οχήματος. Η επίδειξη αύξησε την ευαισθητοποίηση σχετικά με τους πιθανούς κινδύνους ασφαλείας των συνδεδεμένων οχημάτων και τη σημασία της ασφαλείας των συσκευών IoT στην αυτοκινητοβιομηχανία.[11]

Stuxnet Worm: Το 2010, ανακαλύφθηκε το σκουλήκι Stuxnet, το οποίο σχεδιάστηκε για να στοχεύει συστήματα βιομηχανικού ελέγχου. Το σκουλήκι στόχευε ειδικά τους προγραμματιζόμενους λογικούς ελεγκτές (PLC) της Siemens που χρησιμοποιούνται σε κρίσιμες υποδομές, όπως σταθμούς ηλεκτροπαραγωγής και εγκαταστάσεις επεξεργασίας νερού. Το σκουλήκι σχεδιάστηκε για να παρεμβαίνει στην κανονική λειτουργία των PLC και να προκαλεί φυσική βλάβη στα συστήματα που έλεγχαν. Το σκουλήκι Stuxnet ήταν αξιοσημείωτο για την πολυπλοκότητα του σχεδιασμού του, καθώς και για το γεγονός ότι ήταν το πρώτο γνωστό παράδειγμα σκουληκιού που σχεδιάστηκε ειδικά για να προκαλεί φυσική βλάβη.[12]

2.3 Ασφάλεια

Οι εταιρείες λαμβάνουν υπόψη την ασφάλεια των ενσωματωμένων συστημάτων από πολύ νωρίς, στη διαδικασία σχεδιασμού, προκειμένου να μειώσουν την έκθεση σε απειλές των συσκευών που δημιουργούν και των δεδομένων που αυτές παράγουν. Πρέπει να μπορούν να εγγυηθούν για την ασφάλεια της ενσωματωμένης συσκευής για ολόκληρο τον κύκλο ζωής της. Οπότε ξεκινάνε πριν ακόμη γραφτεί η πρώτη γραμμή κώδικα, προσθέτουν δικλίδες ασφαλείας σε περίπτωση που κάποιος εισβολέας καταφέρει να πάρει στην κατοχή του μια συσκευή και φροντίζουν όλα τα συστήματα ασφαλείας να διαρκέσουν μέχρι να χαλάσει μια συσκευή.

Μια πολιτική ασφαλείας για ενσωματωμένα συστήματα χρησιμοποιεί την τριάδα CIA ως μοντέλο για την ανάπτυξη της. Η τριάδα CIA ορίζει τις αρχές που απαιτούνται για την προστασία από μη εξουσιοδοτημένη πρόσβαση, χρήση, αποκάλυψη, διακοπή, τροποποίηση ή καταστροφή μιας συσκευής. Αυτό το μοντέλο βοηθά τις ομάδες ανάπτυξης να σκεφτούν τις διάφορες πτυχές ασφαλείας για το προϊόν τους. Το ακρωνύμιο CIA σημαίνει εμπιστευτικότητα (confidentiality), ακεραιότητα (integrity) και διαθεσιμότητα (availability).[13]

- Confidentiality: Η εμπιστευτικότητα αφορά την παρεμπόδιση της πρόσβασης των μη εξουσιοδοτημένων χρηστών σε ευαίσθητες πληροφορίες που είναι αποθηκευμένες στο σύστημα, παράγονται ή κοινοποιούνται από αυτό.
- Integrity: Η ακεραιότητα διασφαλίζει ότι τα δεδομένα στο ενσωματωμένο σύστημα δεν έχουν διαγραφεί ή τροποποιηθεί από κάποιον χωρίς άδεια.
- Availability: Η διαθεσιμότητα αναφέρεται στο ότι το ενσωματωμένο σύστημα είναι προσβάσιμο όταν χρειάζεται και χωρίς αδικαιολόγητη καθυστέρηση, κατόπιν αιτήματος εξουσιοδοτημένου φορέα.

”Ασφάλεια του υπολογιστή είναι η διαδικασία διασφάλισης της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας των υπολογιστών, των προγραμμάτων και των δεδομένων τους. Η έλλειψη ασφαλείας είναι αποτέλεσμα αποτυχίας μιας από αυτές τις τρεις ιδιότητες.”[14]

2.3.1 Μοντέλο απειλών

Ένα μοντέλο απειλών για συσκευές IoT είναι μια δομημένη προσέγγιση για τον εντοπισμό πιθανών απειλών ασφαλείας και τρωτών σημείων που θα μπορούσαν

να εκμεταλλευτούν οι εισβολείς. Περιλαμβάνει τον εντοπισμό πιθανών φορέων επίθεσης, τον προσδιορισμό της πιθανότητας και του αντίκτυπου μιας επιτυχημένης επίθεσης και την ανάπτυξη στρατηγικών για τον μετριασμό ή τη μείωση του κινδύνου.

Κάποιες βασικές έννοιες που πρέπει να ληφθούν υπόψη κατά την ανάπτυξη ενός μοντέλου απειλών για συσκευές IoT είναι:

- Τρωτά σημεία (Vulnerabilities): Πρόκειται για αδυναμίες ή ελαττώματα στη σχεδίαση ή την υλοποίηση μιας συσκευής που μπορούν να εκμεταλλευτούν οι εισβολείς για να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση, να κλέψουν δεδομένα ή να προκαλέσουν άλλους τύπους βλάβης.
- Τρόποι εκμετάλλευσης (Exploits): Πρόκειται για εργαλεία ή τεχνικές που χρησιμοποιούνται από τους εισβολείς για να εκμεταλλευτούν τα τρωτά σημεία σε μια συσκευή ή ένα σύστημα. Οι τρόποι αυτοί μπορούν να κυμαίνονται από απλά script έως εξελιγμένο κακόβουλο λογισμικό που έχει σχεδιαστεί για να παρακάμπτει τους ελέγχους ασφαλείας και να αποκτά πρόσβαση σε ευαίσθητα δεδομένα.
- Επιτιθέμενοι (Attackers): Πρόκειται για άτομα ή ομάδες που επιδιώκουν να εκμεταλλευτούν τα τρωτά σημεία σε συσκευές IoT για δικό τους κέρδος. Μπορεί να υποκινούνται από οικονομικό όφελος, πολιτικά κίνητρα ή άλλους λόγους.
- Απειλές (Threats): Αυτοί είναι δυνητικοί κίνδυνοι ασφάλειας που θα μπορούσαν να προκύψουν από έναν εισβολέα που εκμεταλλεύεται ελαττώματα σε μια συσκευή IoT. Αυτές οι απειλές μπορεί να περιλαμβάνουν κλοπή δεδομένων, μη εξουσιοδοτημένη πρόσβαση, μόλυνση από κακόβουλο λογισμικό και άλλους τύπους βλάβης.

Μερικά τρωτά σημεία, σε συσκευές IoT, τα οποία επιτρέπουν συνήθως επιθέσεις περιλαμβάνουν:

- Αδύναμοι κωδικοί πρόσβασης: Πολλές συσκευές IoT αποστέλλονται με αδύναμους ή προεπιλεγμένους κωδικούς πρόσβασης που μπορούν εύκολα να μαντέψουν ή να εκμεταλλευτούν οι εισβολείς.

- Μη ασφαλή κανάλια επικοινωνίας: Οι συσκευές IoT ενδέχεται να μεταδίδουν δεδομένα μέσω μη ασφαλών καναλιών, διευκολύνοντας τους εισβολείς να υποκλέψουν ή να τροποποιήσουν τα δεδομένα.
- Μη επιδιορθωμένα τρωτά σημεία: Οι συσκευές IoT ενδέχεται να μην ενημερώνονται με τις πιο πρόσφατες ενημερώσεις κώδικα ασφαλείας, με αποτέλεσμα να είναι ευάλωτες σε γνωστές εκμεταλλεύσεις.
- Ανασφαλές υλικολογισμικό: Οι συσκευές IoT μπορεί να διαθέτουν υλικολογισμικό που μπορεί εύκολα να τροποποιηθεί από εισβολείς, επιτρέποντάς τους να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση ή έλεγχο.

Για να αναπτυχθεί ένα μοντέλο απειλής για συσκευές IoT, είναι σημαντικό να εντοπιστούν αυτά και άλλα πιθανά τρωτά σημεία και φορείς επίθεσης και να αξιολογηθεί η πιθανότητα και ο πιθανός αντίκτυπος μιας επιτυχημένης επίθεσης. Αυτό μπορεί να περιλαμβάνει τη διενέργεια αξιολογήσεων κινδύνου, ασκήσεις μοντελοποίησης απειλών και δοκιμές διείσδυσης για τον εντοπισμό αδυναμιών και την ανάπτυξη στρατηγικών για τον μετριασμό τους.

2.3.2 Ταξινόμηση επιθέσεων

Στο ανώτατο επίπεδο, οι επιθέσεις ταξινομούνται σε τρεις κύριες κατηγορίες με βάση τους λειτουργικούς τους στόχους.[15][16]

- Επιθέσεις απορρήτου (Privacy attacks): Ο στόχος αυτών των επιθέσεων είναι η απόκτηση γνώσης ευαίσθητων πληροφοριών που αποθηκεύονται, κοινοποιούνται ή χειρίζονται σε ένα ενσωματωμένο σύστημα.
- Επιθέσεις ακεραιότητας (Integrity attacks): Αυτές οι επιθέσεις επιχειρούν να αλλάξουν δεδομένα ή κώδικα που σχετίζεται με ένα ενσωματωμένο σύστημα.
- Επιθέσεις διαθεσιμότητας (Availability attacks): Αυτές οι επιθέσεις διαταράσσουν την κανονική λειτουργία του συστήματος με την κατάχρηση πόρων του συστήματος έτσι ώστε να μην είναι διαθέσιμοι για κανονική λειτουργία.

Ένα δεύτερο επίπεδο ταξινόμησης των επιθέσεων σε ενσωματωμένα συστήματα βασίζεται στους παράγοντες ή τα μέσα που χρησιμοποιούνται για την πραγματοποίηση των επιθέσεων. Αυτοί συνήθως ομαδοποιούνται σε τρεις κύριες κατηγορίες:

- Επιθέσεις λογισμικού, οι οποίες αναφέρονται σε επιθέσεις που εξαπολύονται μέσω κακόβουλου λογισμικού όπως ιοί, δούρειοι ίπποι, σκουλήκια κλπ.
- Φυσικές ή επεμβατικές επιθέσεις, οι οποίες αναφέρονται σε επιθέσεις που απαιτούν φυσική εισβολή στο σύστημα σε κάποιο επίπεδο (τσιπ, πλακέτα ή επίπεδο συστήματος).
- Επιθέσεις πλευρικού καναλιού, οι οποίες αναφέρονται σε επιθέσεις που βασίζονται στην παρατήρηση των ιδιοτήτων του συστήματος ενώ εκτελεί κρυπτογραφικές λειτουργίες, π.χ. χρόνο εκτέλεσης, κατανάλωση ενέργειας ή συμπεριφορά κατά την παρουσία σφαλμάτων.

Επιθέσεις λογισμικού

Κακόβουλο λογισμικό Μια επίθεση κακόβουλου λογισμικού δεν είναι μια συνηθισμένη επίθεση στα ενσωματωμένα συστήματα διότι οι συσκευές συνήθως δεν είναι προσβάσιμες από το κοινό χρήστη. Οι μόνοι που έχουν πρόσβαση στη συσκευή και μπορούν να εγκαταστήσουν ή να αναβαθμίσουν το λογισμικό είναι οι κατασκευαστές του. Το κακόβουλο λογισμικό (γνωστός και ως ιός) εκτελεί μη εξουσιοδοτημένες ενέργειες στο σύστημα του θύματος και οι στόχοι του ποικίλλουν. Πρώτος στόχος είναι η εξαγωγή πληροφοριών, η κλοπή δεδομένων και διαπιστευτηρίων, από το ενσωματωμένο σύστημα. Το κακόβουλο λογισμικό που επικεντρώνεται σε αυτό το είδος κλοπής μπορεί να είναι εξαιρετικά δαπανηρό και επικίνδυνο για το άτομο ή την εταιρεία που πέφτει θύμα, καθώς υπάρχει πιθανότητα διάρρηξης αυτών των πληροφοριών. Ο δεύτερος στόχος είναι η διατάραξη της λειτουργίας του συστήματος, από έναν ιό σε μια μεμονωμένη συσκευή που καταστρέφει κρίσιμα αρχεία του λειτουργικού συστήματος, καθιστώντας όλο το σύστημα άχρηστο, έως μια οργανωμένη αυτοκαταστροφή πολλών συσκευών στο σύστημα μιας εγκατάστασης, το επίπεδο «διακοπής» μπορεί να ποικίλλει. Υπάρχει, επίσης, το σενάριο όπου μολυσμένα συστήματα πραγματοποιούν επιθέσεις μεγάλης κλίμακας κατανεμημένης άρνησης υπηρεσίας (DDOS) στο ενσωματωμένο σύστημα. Τέλος, ένας ακόμα στόχος για την κατάχρηση του κακόβουλου λογισμικού θα μπορούσε να είναι η άμεση εκβίαση χρημάτων από το θύμα, για παράδειγμα το Scareware χρησιμοποιεί κενές απειλές (αυτές που δεν τεκμηριώνονται ή/και δε θα μπορούσαν πραγματικά να εκτελεστούν) για να «τρομάξει» τον στόχο. Μια άλλη εναλλακτική, το Ransomware είναι ένας τύπος κακόβουλου λογισμικού που προσπαθεί να εμποδίσει έναν στόχο

από το να αποκτήσει πρόσβαση στα δεδομένα του (συνήθως κρυπτογραφώντας αρχεία στον στόχο) έως ότου επιτευχθεί ο απώτερος σκοπός.[17]

Υπερχείλιση μνήμης Μια υπερχείλιση buffer συμβαίνει όταν ένα πρόγραμμα ή μια διεργασία προσπαθεί να γράφει περισσότερα δεδομένα από αυτά που μπορεί να χωρέσει η προσωρινή μνήμη. Όπως υποδηλώνει το όνομα, τα επιπλέον δεδομένα θα υπερχειλίσουν στη μνήμη δίπλα στο buffer και θα αντικαταστήσουν τα δεδομένα που υπάρχουν ήδη εκεί. Οι εισβολείς μπορούν να εκμεταλλευτούν μια υπερχείλιση buffer για να ελέγξουν, να διακόψουν ή να τροποποιήσουν μια διαδικασία του επεξεργαστή. Οι υπερχειλίσεις buffer είναι μία από τις πιο κοινές αδυναμίες λογισμικού και μία από τις πιο επικίνδυνες, μπορεί να συμβούν ακούσια ή να προκληθούν από κακόβουλο παράγοντα, αλλά με κάθε τρόπο δημιουργεί ευκαιρίες για επιθέσεις. Οι τεχνικές επίθεσης χάκερ διαφέρουν μεταξύ των συστημάτων, αλλά συνήθως περιλαμβάνουν παραβίαση γλωσσών προγραμματισμού και υπέρβαση των ορίων των buffer για την ενεργοποίηση πρόσθετων ενεργειών. Αυτές οι ενέργειες θα μπορούσαν να περιλαμβάνουν την αποστολή νέων οδηγιών στον επεξεργαστή, όπως η παροχή πρόσβασης στον χάκερ σε συστήματα πληροφορικής ή η καταστροφή και η κατάληψη του MCU. Οι τύποι υπερχείλισης buffer κατηγοριοποιούνται ανάλογα με τη θέση του buffer στη μνήμη διεργασίας. Παραδείγματα επιθέσεων υπερχείλισης buffer περιλαμβάνουν επιθέσεις με βάση το stack, το heap, ακέραιους αριθμούς, συμβολοσειρές και Unicode overflow επιθέσεις.[18]

Φυσικές επιθέσεις

Οι φυσικές επιθέσεις σε συσκευές IoT περιλαμβάνουν τη χρήση φυσικής βίας ή χειραγώγησης για να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση στη συσκευή ή στα δεδομένα της. Αυτό μπορεί να περιλαμβάνει ενέργειες όπως η παραβίαση της συσκευής, η χρήση εξειδικευμένων εργαλείων για την παράκαμψη μέτρων ασφαλείας (probing) ή ακόμη και η πλήρης καταστροφή της συσκευής. Αυτές οι επιθέσεις μπορεί να έχουν σοβαρές συνέπειες, όπως να επιτρέψουν στους εισβολείς να κλέψουν ευαίσθητες πληροφορίες, να διακόψουν τη λειτουργία της συσκευής ή ακόμη και να προκαλέσουν ζημιά στον κάτοχο της συσκευής.

Ένα παράδειγμα φυσικής επίθεσης σε μια συσκευή IoT είναι το λεγόμενο «σπάσιμο» ενός έξυπνου ηχείου. Σε αυτόν τον τύπο επίθεσης, ένας εισβολέας καταστρέφει τη συσκευή σε μια προσπάθεια να αποκτήσει πρόσβαση στα δεδομένα που είναι

αποθηκευμένα σε αυτήν. Αυτό μπορεί να γίνει για την κλοπή ευαίσθητων πληροφοριών, όπως οικονομικά ή προσωπικά δεδομένα, ή για να διακοπεί η λειτουργία της συσκευής. Ένα άλλο παράδειγμα είναι η χρήση εξειδικευμένων εργαλείων, όπως οι σαρωτές RFID, για την παράκαμψη μέτρων ασφαλείας σε μια έξυπνη κλειδαριά. Αυτό επιτρέπει στον εισβολέα να αποκτήσει μη εξουσιοδοτημένη πρόσβαση στη συσκευή, δίνοντάς του ενδεχομένως πρόσβαση στο σπίτι του ιδιοκτήτη.

Οι πιθανοί κίνδυνοι φυσικών επιθέσεων σε συσκευές IoT είναι πολυάριθμοι και ποικίλοι. Για τα άτομα, αυτό μπορεί να σημαίνει απώλεια προσωπικών πληροφοριών ή διακοπή της καθημερινότητάς τους. Για τις επιχειρήσεις, οι φυσικές επιθέσεις σε συσκευές IoT θα μπορούσαν να οδηγήσουν σε οικονομικές απώλειες, παραβιάσεις δεδομένων ή ακόμα και βλάβη στη φήμη της εταιρείας. Επιπλέον, η άνοδος των ιατρικών συσκευών που συνδέονται με το IoT, όπως οι βηματοδότες και οι αντλίες ινσουλίνης, έχει εγείρει ανησυχίες σχετικά με την πιθανότητα σωματικών επιθέσεων να προκαλέσουν βλάβη σε άτομα.

Επιθέσεις διαταραχών

Οι επιθέσεις διαταραχών περιλαμβάνουν σκόπιμη εισαγωγή σφαλμάτων ή σφαλμάτων στο σύστημα για να προκαλέσει απροσδόκητη συμπεριφορά ή δυσλειτουργία. Αυτός ο τύπος επίθεσης μπορεί να πραγματοποιηθεί με την εκμετάλλευση ευπαθειών στο σύστημα, όπως αυτά που σχετίζονται με την επικύρωση εισόδου ή τον χειρισμό σφαλμάτων.

- Μια κοινή μέθοδος διεξαγωγής επιθέσεων διαταραχής είναι μέσω της fault injection, η οποία περιλαμβάνει σκόπιμη εισαγωγή σφαλμάτων στο σύστημα για να παρατηρηθεί πώς αποκρίνεται. Για παράδειγμα, ένας εισβολέας μπορεί να εισάγει σφάλματα στα δεδομένα εισόδου, στο τροφοδοτικό ή στο σήμα ρολογιού της συσκευής για να δει πώς αντιδρά το σύστημα και να εντοπίσει τυχόν αδυναμίες που μπορούν να εκμεταλλευτούν. Οι επιθέσεις fault injection είναι ιδιαίτερα δύσκολο να αμυνθούν, καθώς μπορεί να είναι δύσκολο να εντοπιστούν και να αναπαραχθούν.

Επιθέσεις πλευρικού καναλιού

Οι επιθέσεις πλευρικού καναλιού είναι ένας τύπος απειλής ασφαλείας που περιλαμβάνει τη χρήση πληροφοριών σχετικά με τη φυσική υλοποίηση μιας συσκευής

για την απόκτηση μη εξουσιοδοτημένης πρόσβασης ή προνομίων. Η εκτέλεση επίθεσης πλευρικού καναλιού σε μια συσκευή IoT μπορεί να είναι μια πολύπλοκη και τεχνική διαδικασία που απαιτεί εξειδικευμένες γνώσεις και εργαλεία. Ακολουθούν ορισμένα γενικά βήματα που μπορεί να εμπλέκονται στη διεξαγωγή μιας τέτοιας επίθεσης:

1. Προσδιορισμός μιας IoT συσκευής στόχου και συλλογή πληροφοριών σχετικά με τη φυσική εφαρμογή και λειτουργία της.
2. Ανάπτυξη ενός μοντέλου των κρυπτογραφικών αλγορίθμων και λειτουργιών της συσκευής στόχου.
3. Χρήση των πληροφοριών που συγκεντρώθηκαν για τη σχεδίαση και εφαρμογή μιας επίθεσης πλευρικού καναλιού.
4. Εκτέλεση της επίθεσης και συγκέντρωση δεδομένων από τη συσκευή στόχο.
5. Αξιοποίηση τις συλλεγμένης αυτής πληροφορίας για μη εξουσιοδοτημένη πρόσβαση στη συσκευή στόχο.

Συνολικά, η διεξαγωγή επίθεσης πλευρικού καναλιού σε μια συσκευή IoT απαιτεί βαθιά κατανόηση της κρυπτογραφίας, της ασφάλειας του υπολογιστή και των φυσικών χαρακτηριστικών της συσκευής στόχου.

Υπάρχουν διάφοροι τύποι επιθέσεων πλευρικών καναλιών που μπορούν να χρησιμοποιηθούν εναντίον συσκευών IoT (Internet of Things). Μερικά κοινά παραδείγματα περιλαμβάνουν:

- Επιθέσεις ανάλυσης ισχύος: Αυτές οι επιθέσεις μετρούν την ποσότητα ηλεκτρικής ενέργειας που καταναλώνεται από μια συσκευή ενώ εκτελεί μια συγκεκριμένη λειτουργία, η οποία μπορεί να αποκαλύψει πληροφορίες για τα δεδομένα που επεξεργάζονται.
- Επιθέσεις χρονισμού: Αυτές οι επιθέσεις εκμεταλλεύονται το χρόνο που χρειάζεται μια συσκευή για να εκτελέσει μια συγκεκριμένη λειτουργία, όπως η κρυπτογράφηση ή η αποκρυπτογράφηση δεδομένων, για να αποκτήσουν πληροφορίες σχετικά με την εσωτερική λειτουργία της συσκευής.

- **Ηλεκτρομαγνητικές επιθέσεις:** Αυτές οι επιθέσεις χρησιμοποιούν ηλεκτρομαγνητική ακτινοβολία, όπως ραδιοκύματα ή μικροκύματα, για να παρεμποδίσουν τη λειτουργία μιας συσκευής ή για να αποκτήσουν πληροφορίες σχετικά με την εσωτερική λειτουργία της.[19]
- **Ακουστικές επιθέσεις:** Αυτές οι επιθέσεις χρησιμοποιούν ηχητικά κύματα για να παρεμποδίσουν τη λειτουργία μιας συσκευής ή για να αποκτήσουν πληροφορίες σχετικά με την εσωτερική λειτουργία της.

Για παράδειγμα, ένας εισβολέας μπορεί να χρησιμοποιήσει μια επίθεση πλευρικού καναλιού για να εξαγάγει τα κρυπτογραφικά κλειδιά που χρησιμοποιούνται από μια συσκευή IoT για την κρυπτογράφηση των επικοινωνιών της. Αυτό θα μπορούσε να γίνει παρατηρώντας προσεκτικά την κατανάλωση ενέργειας της συσκευής ενώ εκτελεί κρυπτογραφικές λειτουργίες και, στη συνέχεια, χρησιμοποιώντας αυτές τις πληροφορίες να αναστρέψει τα κλειδιά. Εναλλακτικά, ένας εισβολέας θα μπορούσε να χρησιμοποιήσει τις ηλεκτρομαγνητικές εκπομπές μιας συσκευής για να συναγάγει τις ίδιες πληροφορίες. Αυτές οι επιθέσεις μπορεί να είναι δύσκολο να εντοπιστούν και να αποφευχθούν και μπορεί να αποτελέσουν σοβαρή απειλή για την ασφάλεια των συστημάτων IoT.[20]

2.3.3 Αντίμετρα

Βέλτιστες πρακτικές

Ασφαλείς πρακτικές ανάπτυξης Αυτό είναι ένα ουσιαστικό πρώτο βήμα για την ασφάλεια των ενσωματωμένων συστημάτων. Ακολουθώντας τις οδηγίες ασφαλούς ανάπτυξης λογισμικού (MISRA-C/C++) και χρησιμοποιώντας ασφαλή πλαίσια, βιβλιοθήκες και API, οι προγραμματιστές μπορούν να διασφαλίσουν ότι τα συστήματα που δημιουργούν είναι ανθεκτικά σε κοινά τρωτά σημεία ασφαλείας.

Έλεγχος ταυτότητας και έλεγχος πρόσβασης Αυτό είναι ένα βασικό στοιχείο της ασφάλειας, καθώς διασφαλίζει ότι μόνο εξουσιοδοτημένοι χρήστες μπορούν να έχουν πρόσβαση στο σύστημα και να εκτελούν ενέργειες σε αυτό. Με την εφαρμογή ισχυρών μηχανισμών ελέγχου ταυτότητας και ελέγχου πρόσβασης, οι οργανισμοί μπορούν να αποτρέψουν τη μη εξουσιοδοτημένη πρόσβαση και να προστατευτούν από πιθανές απειλές.

Κρυπτογράφηση δεδομένων Η κρυπτογράφηση δεδομένων είναι ένα σημαντικό μέτρο ασφαλείας, καθώς προστατεύει ευαίσθητες πληροφορίες από υποκλοπή ή κλοπή από μη εξουσιοδοτημένους χρήστες. Με την κρυπτογράφηση δεδομένων κατά τη μεταφορά αλλά και όσο αυτά είναι αποθηκευμένα, οι οργανισμοί μπορούν να διασφαλίσουν ότι οι ευαίσθητες πληροφορίες τους παραμένουν προστατευμένες, ακόμα και αν πέσουν σε λάθος χέρια.

Ασφαλής εκκίνηση Μια ασφαλής διαδικασία εκκίνησης είναι επίσης ουσιαστικό μέρος της προστασίας ενός ενσωματωμένου συστήματος από κακόβουλο λογισμικό. Διασφαλίζοντας ότι μόνο αξιόπιστο λογισμικό μπορεί να εκτελεστεί στο σύστημα, οι οργανισμοί μπορούν να αποτρέψουν τους εισβολείς από το να υπονομεύσουν τα συστήματά τους και να κλέψουν ευαίσθητα δεδομένα.

Παρακολούθηση και καταγραφή δραστηριότητας Η παρακολούθηση και η καταγραφή της δραστηριότητας του συστήματος είναι ένα σημαντικό μέρος του εντοπισμού και της απόκρισης σε συμβάντα ασφαλείας. Με την τακτική παρακολούθηση και καταγραφή της δραστηριότητας του συστήματος, οι οργανισμοί μπορούν να εντοπίσουν πιθανές ευπάθειες ασφαλείας και να αναλάβουν δράση για την αντιμετώπισή τους προτού μπορέσουν να γίνουν αντικείμενο εκμετάλλευσης από εισβολείς.

Προστασία από φυσικές επιθέσεις Τα ενσωματωμένα συστήματα είναι συχνά μικρά και φορητά, καθιστώντας τα ευάλωτα σε φυσικές επιθέσεις. Για την προστασία από αυτούς τους τύπους απειλών, οι οργανισμοί θα πρέπει να εφαρμόζουν μέτρα όπως ασφαλή περιβλήματα, υλικό ανθεκτικό σε παραβιάσεις και ελέγχους πρόσβασης για την αποτροπή φυσικής πρόσβασης στο σύστημα.

Εκπαίδευση των χρηστών Πολλές παραβιάσεις ασφάλειας στο IoT και στα ενσωματωμένα συστήματα συμβαίνουν επειδή οι χρήστες δε γνωρίζουν τις βέλτιστες πρακτικές για τη διατήρηση της ασφάλειας των συσκευών τους. Είναι σημαντικό να εκπαιδεύονται οι χρήστες σχετικά με τη σημασία της χρήσης ισχυρών κωδικών πρόσβασης, τη διατήρηση ενημερωμένου λογισμικού και υλικολογισμικού και άλλα μέτρα ασφαλείας.

Απομόνωση βάσει λογισμικού

Η "απομόνωση βάσει λογισμικού" περιλαμβάνει τη χρήση λογισμικού για τη δημιουργία ενός ασφαλούς περιβάλλοντος εντός του οποίου μπορεί να λειτουργήσει μια συσκευή IoT. Αυτό συνήθως επιτυγχάνεται μέσω της χρήσης της τεχνολογίας εικονικοποίησης, η οποία επιτρέπει την ύπαρξη πολλαπλών απομονωμένων περιβαλλόντων σε μια ενιαία φυσική συσκευή. Στην περίπτωση συσκευών IoT, αυτό μπορεί να περιλαμβάνει τη δημιουργία ενός εικονικού περιβάλλοντος για κάθε μεμονωμένη συσκευή, με αυστηρούς ελέγχους για την αποτροπή μη εξουσιοδοτημένης πρόσβασης ή επικοινωνίας από εξωτερικές πηγές.

Ένα από τα βασικά οφέλη της απομόνωσης βάσει λογισμικού είναι ότι επιτρέπει στους κατασκευαστές να προστατεύουν τις συσκευές IoT χωρίς να προσθέτουν σημαντική πολυπλοκότητα υλικού ή κόστος. Χρησιμοποιώντας την τεχνολογία εικονικοποίησης, οι κατασκευαστές μπορούν να δημιουργήσουν ασφαλή περιβάλλοντα σε υπάρχον υλικό, χωρίς την ανάγκη πρόσθετων στοιχείων υλικού ή εξειδικευμένων συσκευών ασφαλείας. Αυτό το καθιστά ελκυστική επιλογή για τους κατασκευαστές που θέλουν να εξασφαλίσουν μεγάλο αριθμό συσκευών IoT, όπως σε περιβάλλον έξυπνου σπιτιού ή συνδεδεμένου γραφείου.

Εκτός από τα πλεονεκτήματα κόστους και πολυπλοκότητας, η απομόνωση βάσει λογισμικού προσφέρει επίσης υψηλό επίπεδο ασφάλειας. Δημιουργώντας ένα ξεχωριστό, απομονωμένο περιβάλλον για κάθε συσκευή, οι κατασκευαστές μπορούν να διασφαλίσουν ότι κάθε συσκευή λειτουργεί ανεξάρτητα και δεν μπορεί να προσπελαστεί ή να επηρεαστεί από άλλες συσκευές στο δίκτυο. Αυτό μπορεί να βοηθήσει στην αποτροπή της εξάπλωσης κακόβουλου λογισμικού ή άλλων απειλών ασφαλείας και μπορεί επίσης να αποτρέψει τη χρήση συσκευών ως μέρος μιας μεγαλύτερης επίθεσης στο δίκτυο.

Απομόνωση βάσει υλικού

Η απομόνωση βάσει υλικού είναι μια άλλη προσέγγιση που μπορεί να χρησιμοποιηθεί για την ασφάλεια συσκευών IoT. Αυτή η προσέγγιση περιλαμβάνει τη χρήση εξειδικευμένων στοιχείων υλικού, όπως τείχη προστασίας υλικού ή μονάδες ασφαλείας υλικού, για τη δημιουργία ενός ασφαλούς περιβάλλοντος για τη λειτουργία της συσκευής.

Ένα από τα βασικά πλεονεκτήματα της απομόνωσης που βασίζεται σε υλικό εί-

ναι ότι μπορεί να παρέχει υψηλότερο επίπεδο ασφάλειας από την απομόνωση που βασίζεται σε λογισμικό. Επειδή τα μέτρα ασφαλείας εφαρμόζονται σε ειδικά εξαρτήματα υλικού, είναι πιο δύσκολο για τους εισβολείς να τα παρακάμψουν ή να τα απενεργοποιήσουν. Αυτό μπορεί να καταστήσει πιο δύσκολο για τους εγκληματίες να αποκτήσουν πρόσβαση σε μια συσκευή ή να χρησιμοποιήσουν τη συσκευή ως μέρος μιας μεγαλύτερης επίθεσης σε ένα δίκτυο.

Ένα άλλο πλεονέκτημα της απομόνωσης που βασίζεται σε υλικό είναι ότι μπορεί να είναι πιο επεκτάσιμη από την απομόνωση που βασίζεται σε λογισμικό. Σε ορισμένες περιπτώσεις, η απομόνωση που βασίζεται σε λογισμικό μπορεί να απαιτεί ένα ξεχωριστό εικονικό περιβάλλον για κάθε μεμονωμένη συσκευή, το οποίο μπορεί να καταστεί μη πρακτικό καθώς αυξάνεται ο αριθμός των συσκευών. Αντίθετα, η απομόνωση που βασίζεται σε υλικό μπορεί συχνά να εφαρμοστεί σε μεγαλύτερη κλίμακα, με ένα μόνο στοιχείο υλικού να παρέχει ασφάλεια για πολλές συσκευές.

Ωστόσο, η απομόνωση που βασίζεται σε υλικό έχει επίσης ορισμένα μειονεκτήματα. Ένα από τα βασικά μειονεκτήματα είναι ότι μπορεί να είναι πιο ακριβό και πολύπλοκο από την απομόνωση που βασίζεται σε λογισμικό. Εξειδικευμένα στοιχεία υλικού, όπως τείχη προστασίας υλικού ή μονάδες ασφαλείας, μπορεί να έχουν υψηλό κόστος αγοράς και συντήρησης και ενδέχεται να απαιτούν πρόσθετη τεχνογνωσία για τη διαμόρφωση και τη διαχείριση τους. Αυτό μπορεί να κάνει την απομόνωση που βασίζεται σε υλικό λιγότερο ελκυστική για τους κατασκευαστές που θέλουν να εξασφαλίσουν μεγάλο αριθμό συσκευών IoT ή για εκείνους που λειτουργούν με περιορισμένο προϋπολογισμό.

Συνολικά, ενώ η απομόνωση που βασίζεται σε υλικό μπορεί να παρέχει υψηλότερο επίπεδο ασφάλειας από την απομόνωση που βασίζεται σε λογισμικό, μπορεί επίσης να είναι πιο δαπανηρή και πολύπλοκη στην εφαρμογή της. Οι κατασκευαστές πρέπει να λαμβάνουν προσεκτικά υπόψη τις συγκεκριμένες ανάγκες ασφαλείας και τους περιορισμούς τους όταν αποφασίζουν εάν θα χρησιμοποιήσουν απομόνωση βάσει υλικού ή λογισμικού για την ασφάλεια των συσκευών IoT τους.

2.4 Ασφάλεια με βάση το υλικό

2.4.1 Root of Trust

Το Root of Trust αναφέρεται σε μια ασφαλή βάση ή σημείο εκκίνησης που μπορεί να χρησιμοποιηθεί για να επαληθευτεί η ακεραιότητα μιας συσκευής ή συστήματος. Είναι ένα σύνολο στοιχείων υλικού ή λογισμικού που έχουν σχεδιαστεί για να παρέχουν μια ασφαλή βάση για τη λειτουργία μιας συσκευής ή συστήματος. Το Root of Trust χρησιμοποιείται συνήθως για την επαλήθευση της αυθεντικότητας και της ακεραιότητας της διαδικασίας εκκίνησης, καθώς και του λογισμικού που εκτελείτε στη συσκευή.

Το Root of Trust μπορεί να χρησιμοποιηθεί για τη δημιουργία μιας αλυσίδας εμπιστοσύνης, στην οποία η ακεραιότητα κάθε επόμενου στοιχείου ή διαδικασίας επαληθεύεται με βάση την αξιοπιστία όλης της αλυσίδας. Αυτό διασφαλίζει ότι η συσκευή ή το σύστημα λειτουργεί σε ασφαλή και αξιόπιστη κατάσταση, ακόμα κι αν έχει παραβιαστεί από κακόβουλο λογισμικό.

Γενικά, το Root of Trust είναι μια σημαντική έννοια στον τομέα της ασφάλειας, καθώς παρέχει έναν τρόπο επαλήθευσης της αυθεντικότητας και της ακεραιότητας μιας συσκευής ή συστήματος και για να διασφαλιστεί ότι λειτουργεί σε ασφαλή και αξιόπιστη κατάσταση.[21]

2.4.2 Platform Security Architecture (PSA)

Η πιστοποίηση Platform Security Architecture (PSA) είναι μια διαδικασία που επαληθεύει την ασφάλεια μιας συσκευής, μιας πλατφόρμας ή ενός συστήματος χρησιμοποιώντας το PSA security framework. Το PSA framework είναι ένα σύνολο οδηγιών και βέλτιστων πρακτικών για την κατασκευή ασφαλών ενσωματωμένων συσκευών και συστημάτων IoT. Το framework αναπτύχθηκε από το PSA Certified Program, το οποίο είναι μια συνεργασία μεταξύ των κορυφαίων κατασκευαστών ημιαγωγών, των πωλητών εργαλείων και των ειδικών σε θέματα ασφάλειας (Μέσα στους οποίους ανήκει και η ARM).

Η διαδικασία πιστοποίησης PSA περιλαμβάνει μια ενδεδειγμένη αξιολόγηση ασφαλείας της συσκευής ή της πλατφόρμας, συμπεριλαμβανομένης της ανασκόπησης του σχεδιασμού υλικού και λογισμικού, των χαρακτηριστικών ασφαλείας και της εφαρμογής. Ο στόχος της πιστοποίησης είναι να παρέχει μια ανεξάρτητη αξιολόγηση

της ασφάλειας της συσκευής και να διασφαλίσει ότι προστατεύεται από κοινές απειλές ασφαλείας, όπως πειρατεία, κακόβουλο λογισμικό και μη εξουσιοδοτημένη πρόσβαση.

Η πιστοποίηση PSA μπορεί να είναι επωφελής για διάφορους ενδιαφερόμενους φορείς, συμπεριλαμβανομένων των κατασκευαστών συσκευών, των προγραμματιστών λογισμικού και των τελικών χρηστών. Για τους κατασκευαστές, η πιστοποίηση PSA παρέχει ένα ανταγωνιστικό πλεονέκτημα επιδεικνύοντας την ασφάλεια των συσκευών και των πλατφορμών τους. Για τους προγραμματιστές, μπορεί να βοηθήσει να διασφαλιστεί ότι το λογισμικό τους είναι ασφαλές και πληροί τα πρότυπα του κλάδου. Και για τους τελικούς χρήστες, η πιστοποίηση PSA παρέχει διαβεβαίωση ότι οι συσκευές και τα συστήματα που χρησιμοποιούν είναι ασφαλή και προστατευμένα από απειλές ασφαλείας.

PSA-RoT

Το PSA-RoT είναι ένας συγκεκριμένος τύπος RoT που έχει σχεδιαστεί και υλοποιηθεί σύμφωνα με τις απαιτήσεις ασφαλείας που ορίζονται στο πλαίσιο Platform Security Architecture (PSA). Το PSA-RoT παρέχει μια ασφαλή βάση για τη συσκευή ή το σύστημα και διασφαλίζει ότι τα χαρακτηριστικά ασφαλείας και η λειτουργικότητα της συσκευής είναι προστατευμένα και αξιόπιστα.

Σε μια συσκευή ή πλατφόρμα με πιστοποίηση PSA, το PSA-RoT είναι υπεύθυνο για την υλοποίηση βασικών λειτουργιών ασφαλείας, όπως ασφαλή εκκίνηση, επικύρωση υλικολογισμικού, κρυπτογραφικές λειτουργίες και ασφαλή αποθήκευση. Εφαρμόζοντας ένα PSA-RoT, οι κατασκευαστές συσκευών μπορούν να βοηθήσουν να διασφαλίσουν ότι οι συσκευές τους είναι ασφαλείς και προστατεύονται από κοινές απειλές ασφαλείας, όπως πειρατεία, κακόβουλο λογισμικό και μη εξουσιοδοτημένη πρόσβαση.[22]

Λειτουργία Ασφαλείας	Περιγραφή
Ασφαλής εκκίνηση	Η διαδικασία επαλήθευσης ότι ο κώδικας που εκτελείται κατά την εκκίνηση είναι νόμιμος και δεν έχει παραβιαστεί.
Κρυπτογραφικές Υπηρεσίες	Οι λειτουργίες που παρέχουν κρυπτογράφηση και αποκρυπτογράφηση δεδομένων, καθώς και δημιουργία και επαλήθευση ψηφιακών υπογραφών.

Λειτουργία Ασφαλείας	Περιγραφή
Ασφαλής αποθήκευση	Η λειτουργία που παρέχει ασφαλή αποθήκευση ευαίσθητων δεδομένων όπως κλειδιά και πιστοποιητικά.
Ταυτότητα συσκευής	Η λειτουργία που παρέχει μια μοναδική και αμετάβλητη ταυτότητα για τη συσκευή που μπορεί να χρησιμοποιηθεί για έλεγχο ταυτότητας και πιστοποίηση.
Ενημέρωση υλικολογισμικού	Η λειτουργία που επιτρέπει στη συσκευή να ενημερώνει με ασφάλεια το υλικολογισμικό της με εξουσιοδοτημένο και επαληθευμένο κωδικό.
Attestation	Η λειτουργία που επιτρέπει στη συσκευή να αποδείξει την ταυτότητα και την ακεραιότητά της σε άλλα μέρη παρέχοντας αποδεικτικά στοιχεία του RoT της και μετρήσεις της κατάστασης του λογισμικού της.
Αρχεία καταγραφής ελέγχου	Η λειτουργία που καταγράφει συμβάντα και ενέργειες που σχετίζονται με την ασφάλεια στη συσκευή για σκοπούς παρακολούθησης και ανάλυσης.

Πίνακας 2.2: Λειτουργίες ασφαλείας PSA Root of Trust

PSA Levels

Το PSA παρέχει τρία επίπεδα σταδιακά αυξανόμενης στιβαρότητας και διασφάλισης ασφάλειας. Το πρώτο επίπεδο πιστοποίησης επιτυγχάνεται μέσω της εξέτασης του documentation και μιας συνέντευξης με ένα από τα εργαστήρια πιστοποίησης. Το δεύτερο επίπεδο περιλαμβάνει εργαστηριακό έλεγχο έναντι της προστασίας PSA-RoT, που σημαίνει ότι το υλικό πρέπει να υποστηρίζει τις λειτουργίες PSA-RoT. Και, το τρίτο επίπεδο υποστηρίζει δοκιμές ενάντια σε πιο επιθετικές και εκλεπτυσμένες επιθέσεις, όπως εισβολές στα πλευρικά κανάλια και φυσική παραβίαση, επομένως, έχει μεγαλύτερη περίοδο αξιολόγησης. Το δεύτερο και το τρίτο επίπεδο απευθύνονται μόνο σε πωλητές chip. Πιο αναλυτικά:

	Level 1	Level 2	Level 3
Διάρκεια	Λίγες εβδομάδες	Λίγους μήνες	Μερικούς μήνες έως ένα χρόνο
Πώς επιτυγχάνεται	Απάντηση σε ερωτηματολόγιο, παροχή documentation και συνέντευξη με εργαστήριο πιστοποίησης	Εφαρμογή χαρακτηριστικών ασφαλείας σύμφωνα με τις οδηγίες του PSA και αξιολόγηση πιστοποιημένου εργαστηρίου	Εφαρμογή χαρακτηριστικών ασφαλείας σύμφωνα με τις οδηγίες του PSA και αξιολόγηση πιστοποιημένου εργαστηρίου
Tests που είναι απαραίτητα	Δε χρειάζονται tests	Επιθέσεις λογισμικού (π.χ. υπερχείλιση buffer 2.3.2)	Επιθέσεις λογισμικού (π.χ. υπερχείλιση buffer 2.3.2), επιθέσεις πλευρικού καναλιού (π.χ. ανάλυση ισχύος 2.3.2), επιθέσεις διαταραχής (π.χ. προκάλεση σφάλματος 2.3.2), φυσικές επιθέσεις (π.χ. probing 2.3.2)
Ποιες δυνατότητες και οφέλη προσφέρονται	Παρέχει διαβεβαίωση ότι η ασφάλεια έχει ληφθεί υπόψη στο στάδιο του σχεδιασμού. Χτίζει εμπιστοσύνη με τους πελάτες.	Παρέχει προστασία από επεκτάσιμες επιθέσεις λογισμικού. Επιδεικνύει στιβαρότητα της εφαρμογής ασφαλείας. Διαφοροποίηση στην αγορά.	Παρέχει προστασία από εξελιγμένες επιθέσεις υλικού και λογισμικού. Επιδεικνύει υψηλό επίπεδο διασφάλισης ασφαλείας. Premium τοποθέτηση στην αγορά.

Πίνακας 2.3: Διαφορές ανάμεσα στα PSA levels

Το PSA είναι μια παγκόσμια συνεργασία που επικεντρώνεται στην πιστοποί-

ηση της IoT ασφάλειας. Η διαδικασία πιστοποίησης περιλαμβάνει τέσσερα στάδια: ανάλυση, αρχιτεκτονική, υλοποίηση και πιστοποίηση. Χρησιμοποιεί κοινά μοντέλα απειλών που βασίζονται σε περιπτώσεις χρήσης IoT, όπως έξυπνες οικιακές συσκευές και βιομηχανικοί αισθητήρες. Το PSA Certified προσφέρει επίσης πρακτικά API για ασφαλείς υπηρεσίες εκκίνησης, υπηρεσίες κρυπτογραφίας και άλλες πτυχές ασφάλειας, τόσο υλικού όσο και λογισμικού, συμπεριλαμβανομένων τεχνικών απομόνωσης και μεθόδων ασφαλούς αποθήκευσης. Ο απώτερος στόχος του PSA Certified είναι η προώθηση της συνέπειας σε ολόκληρο τον κλάδο και η διαφάνεια στην αγορά.

2.4.3 Trusted execution environment

Ένα trusted execution environment (TEE) είναι μια ασφαλής περιοχή του υλικού μιας συσκευής που παρέχει έναν προστατευμένο χώρο για την εκτέλεση ευαίσθητων εφαρμογών και την αποθήκευση ευαίσθητων δεδομένων. Αυτό επιτρέπει στη συσκευή να διατηρεί αυτές τις πληροφορίες ασφαλείς από εξωτερικές απειλές, όπως κακόβουλο λογισμικό ή χάκερ, ενώ εξακολουθεί να επιτρέπει την πρόσβαση σε αυτές από εξουσιοδοτημένες εφαρμογές.

Ένα από τα βασικά χαρακτηριστικά ενός TEE είναι ότι είναι απομονωμένο από το υπόλοιπο λειτουργικό σύστημα της συσκευής. Αυτό σημαίνει ότι δεν είναι προσβάσιμο στον χρήστη ή σε άλλες εφαρμογές που εκτελούνται στη συσκευή, παρέχοντας ένα επιπλέον επίπεδο ασφάλειας. Επιπλέον, τα TEE είναι συχνά κρυπτογραφημένα, προστατεύοντας περαιτέρω τις ευαίσθητες πληροφορίες που περιέχουν.

Τα TEE χρησιμοποιούνται συνήθως σε εφαρμογές που απαιτούν υψηλό επίπεδο ασφάλειας, όπως mobile banking, διαχείριση ψηφιακών δικαιωμάτων και συστήματα πληρωμών. Για παράδειγμα, μια εφαρμογή mobile banking μπορεί να χρησιμοποιήσει ένα TEE για να αποθηκεύσει με ασφάλεια τα διαπιστευτήρια σύνδεσης ενός χρήστη και άλλες ευαίσθητες οικονομικές πληροφορίες, όπως αριθμούς λογαριασμών ή ιστορικό συναλλαγών. Αυτό διασφαλίζει ότι αυτές οι πληροφορίες προστατεύονται από εξωτερικές απειλές και είναι προσβάσιμες μόνο από την εξουσιοδοτημένη εφαρμογή mobile banking.

Συνολικά, τα TEE διαδραματίζουν κρίσιμο ρόλο στη διασφάλιση της ασφάλειας ευαίσθητων πληροφοριών και εφαρμογών στις συσκευές. Καθώς ο όγκος των ευαίσθητων δεδομένων που αποθηκεύονται και επεξεργάζονται στις συσκευές μας συ-

νεχίζει να αυξάνεται, η χρήση των TEE θα γίνει ακόμη πιο σημαντική για τη διασφάλιση ότι αυτές οι πληροφορίες διατηρούνται ασφαλείς από εξωτερικές απειλές.

Μερικά παραδείγματα αξιόπιστων περιβαλλόντων εκτέλεσης (TEE) περιλαμβάνουν το Ασφαλές περιβάλλον εκτέλεσης της Qualcomm (QSEE) και το TrustZone της ARM.

Το Qualcomm Secure Execution Environment (QSEE) είναι ένα TEE που αναπτύχθηκε από την Qualcomm για χρήση στους επεξεργαστές της για κινητά. Παρέχει έναν ασφαλή χώρο για την εκτέλεση ευαίσθητων εφαρμογών και την αποθήκευση ευαίσθητων δεδομένων και είναι απομονωμένος από το υπόλοιπο λειτουργικό σύστημα της συσκευής.

Το ARM TrustZone είναι ένα άλλο παράδειγμα TEE. Είναι μια επέκταση ασφαλείας στην αρχιτεκτονική ARM που παρέχει επίσης ένα ασφαλές περιβάλλον για την εκτέλεση ευαίσθητων εφαρμογών και την αποθήκευση ευαίσθητων δεδομένων. Όπως και το QSEE, είναι απομονωμένο από το υπόλοιπο λειτουργικό σύστημα της συσκευής και παρέχει ένα επιπλέον επίπεδο ασφάλειας για ευαίσθητες πληροφορίες.

2.4.4 TrustZone απο την ARM

Το TrustZone είναι ένα χαρακτηριστικό ασφαλείας ενσωματωμένο σε ορισμένες CPU που βασίζονται σε ARM που παρέχει ένα ασφαλές περιβάλλον εκτέλεσης για ευαίσθητες λειτουργίες. Ο σκοπός του TrustZone είναι να προστατεύει από ορισμένους τύπους επιθέσεων, όπως tampering, χρησιμοποιώντας μέτρα ασφαλείας που βασίζονται στο υλικό.

Όταν μια CPU με TrustZone είναι ενεργοποιημένη, εισέρχεται σε έναν "ασφαλή κόσμο" όπου εκτελούνται ευαίσθητες λειτουργίες. Αυτός ο ασφαλής κόσμος είναι απομονωμένος από το υπόλοιπο σύστημα και έχει τη δική του ξεχωριστή μνήμη και περιφερειακά. Αυτή η απομόνωση διασφαλίζει ότι δεν είναι δυνατή η πρόσβαση ή η παραβίαση των ευαίσθητων λειτουργιών από κακόβουλο λογισμικό ή άλλες απειλές ασφαλείας.

Το Arm TrustZone υλοποιείται χρησιμοποιώντας μια προσέγγιση βασισμένη στο υλικό, πράγμα που σημαίνει ότι είναι ενσωματωμένο στο SoC σε επίπεδο υλικού. Αυτό παρέχει μια σειρά από πλεονεκτήματα, όπως βελτιωμένη απόδοση και ασφάλεια, καθώς και καλύτερη υποστήριξη για λειτουργίες σε πραγματικό χρόνο. Επι-

πλέον, επειδή το TrustZone υλοποιείται σε επίπεδο υλικού, είναι δύσκολο για τους εισβολείς να το παρακάμψουν ή να το απενεργοποιήσουν, καθιστώντας το μια πιο ασφαλή επιλογή από τις λύσεις ασφαλείας που βασίζονται σε λογισμικό.

Ο Ασφαλής κόσμος έχει τη δική μνήμη, καταχωρητές και περιφερειακά που είναι απομονωμένα από τον μη ασφαλή κόσμο. Ο μη ασφαλής κόσμος έχει πρόσβαση σε ένα υποσύνολο της μνήμης, των καταχωρητών και των περιφερειακών του επεξεργαστή. Οι ασφαλείς και μη ασφαλείς περιοχές της μνήμης είναι φυσικά διαχωρισμένες και διαθέτουν τους δικούς τους μηχανισμούς προστασίας για την αποτροπή μη εξουσιοδοτημένης πρόσβασης.

Ένα από τα βασικά πλεονεκτήματα του TrustZone είναι ότι επιτρέπει τη δημιουργία "αξιόπιστων" εφαρμογών, οι οποίες είναι εφαρμογές που έχουν σχεδιαστεί για να εκτελούνται στον ασφαλή κόσμο και να εκτελούν ευαίσθητες λειτουργίες. Αυτές οι εφαρμογές επαληθεύονται από το υλικό για να διασφαλιστεί ότι είναι αξιόπιστες και τους παρέχεται πρόσβαση στον ασφαλή κόσμο και τους πόρους του.

Μια άλλη βασική πτυχή του Arm TrustZone είναι ότι παρέχει μια τυποποιημένη προσέγγιση στην ασφάλεια για συσκευές που βασίζονται σε ARM. Αυτό σημαίνει ότι οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές που είναι συμβατές με το TrustZone και μπορούν εύκολα να ενσωματωθούν σε ένα ευρύ φάσμα συσκευών. Αυτή η τυποποίηση διευκολύνει επίσης τους κατασκευαστές συσκευών να ενσωματώσουν το TrustZone στα προϊόντα τους, παρέχοντας μια πιο συνεπή και ασφαλή εμπειρία χρήστη σε διαφορετικές συσκευές.

Συνοπτικά, το Arm TrustZone είναι μια τεχνολογία ασφαλείας που παρέχει ένα ασφαλές περιβάλλον σε μια συσκευή που λειτουργεί σε επεξεργαστή που βασίζεται σε ARM. Εφαρμόζεται σε επίπεδο υλικού και παρέχει μια τυποποιημένη προσέγγιση για την ασφάλεια, καθιστώντας το ένα σημαντικό εργαλείο για την προστασία ευαίσθητων δεδομένων και λειτουργιών σε συσκευές που βασίζονται σε βραχίονα.

Πως διαμερίζεται η μνήμη σε συμβατικό CPU

Μια μονάδα προστασίας μνήμης (MPU) είναι ένα στοιχείο υλικού της κεντρικής μονάδας επεξεργασίας (CPU) ενός υπολογιστή που παρέχει προστασία μνήμης παρακολουθώντας την πρόσβαση στη μνήμη μέσω διαφορετικών διεργασιών και διασφαλίζοντας ότι κάθε διεργασία μπορεί να έχει πρόσβαση μόνο στη μνήμη στην οποία είναι εξουσιοδοτημένη να έχει πρόσβαση. Αυτό γίνεται συνήθως με τη διαίρεση της μνήμης σε διαφορετικές περιοχές, που ονομάζονται "περιοχές μνήμης" και

ορίζοντας δικαιώματα για κάθε περιοχή που καθορίζουν ποιες διεργασίες μπορούν να έχουν πρόσβαση σε αυτήν την περιοχή. Η MPU ελέγχει τα δικαιώματα για κάθε πρόσβαση στη μνήμη για να διασφαλίσει ότι η διαδικασία που προσπαθεί να αποκτήσει πρόσβαση στη μνήμη είναι εξουσιοδοτημένη να το κάνει. Εάν η διεργασία δεν έχει τα απαραίτητα δικαιώματα, η MPU δημιουργεί exception, η οποία αναγκάζει τη CPU να σταματήσει τη διαδικασία και να μεταφέρει τον έλεγχο σε μια ρουτίνα χειρισμού σφαλμάτων. Αυτό βοηθά στην αποτροπή της πρόσβασης κακόβουλων ή σφαλμάτων διεργασιών στη μνήμη στην οποία δεν υποτίθεται ότι έχουν πρόσβαση, γεγονός που μπορεί να προκαλέσει σφάλματα ή παραβιάσεις ασφάλειας.[23]

Πως διαμερίζεται η μνήμη με TrustZone

SAU στο TrustZone σημαίνει "Security Attribution Unit" και IDAU "Implementation defined Attribute Unit" και είναι παρόμοιες με μια μονάδα προστασίας μνήμης (MPU) καθώς παρέχουν προστασία μνήμης παρακολουθώντας την πρόσβαση στη μνήμη μέσω διαφορετικών διεργασιών και διασφαλίζοντας ότι κάθε διεργασία μπορεί να έχει πρόσβαση μόνο στη μνήμη που έχει εξουσιοδότηση πρόσβασης. Η κύρια διαφορά είναι ότι μια SAU έχει σχεδιαστεί ειδικά για χρήση σε συνδυασμό με το TrustZone. Η SAU μπορεί να επιβάλει διαφορετικά δικαιώματα πρόσβασης στη μνήμη για διαφορετικές καταστάσεις ασφαλείας, γεγονός που επιτρέπει πιο λεπτομερή έλεγχο της πρόσβασης στη μνήμη και καλύτερη προστασία από απειλές ασφαλείας. Είναι δυνατόν οι CPU που βασίζονται σε ARM που χρησιμοποιούν τεχνολογία TrustZone να διαθέτουν MPU, αλλά δεν απαιτείται. Το αν μια ARM CPU που χρησιμοποιεί TrustZone περιλαμβάνει ή όχι μια MPU εξαρτάται από τη συγκεκριμένη υλοποίηση της CPU.

Η IDAU είναι μια μονάδα υλικού που βρίσκεται έξω από τον επεξεργαστή, ενώ η SAU είναι προγραμματιζόμενη μονάδα λογισμικού που βρίσκεται μέσα στον επεξεργαστή. Και τα δύο μπορούν να χρησιμοποιηθούν για τη διαμερισμό της μνήμης του συστήματος σε ασφαλείς/μη ασφαλείς περιοχές.

Η SAU είναι εξ ολοκλήρου μια εσωτερική μονάδα και είναι προγραμματιζόμενη από λογισμικό, η IDAU είναι μια εξωτερική μονάδα αλλά έχει ένα εσωτερικό αντίστοιχο μέσα στον επεξεργαστή, το οποίο συνδέεται με το εξωτερικό IDAU μέσω της διεπαφής IDAU του επεξεργαστή. Το IDAU είναι συνήθως ένας σταθερός μηχανισμός για την κατανομή της μνήμης σε ασφαλείς/μη ασφαλείς περιοχές. Σε περίπτωση που το IDAU δε χρησιμοποιείται, τα σήματα διασύνδεσης στον επεξεργαστή,

πρέπει να συνδέονται με μια συνιστώμενη τιμή όπως ορίζεται στο Εγχειρίδιο ενοποίησης και υλοποίησης. Η SAU δεν είναι σταθερός μηχανισμός, και σε αντίθεση με την IDAU, είναι πλήρως προγραμματιζόμενη από λογισμικό που εκτελείται με «ασφαλή» δικαιώματα πρόσβασης.

```
/* Initialize and enable the SAU */
#define SAU_INIT_CTRL      1
#define SAU_INIT_CTRL_ENABLE  1
...
/* <e>Initialize SAU Region 1 with memory attributes */
#define SAU_INIT_REGION1    1
#define SAU_INIT_START1     0x08040000 /* start address of SAU region 1 */
#define SAU_INIT_END1       0x0807FFFF /* end address of SAU region 1 */
/* Region can be set as: 0 = non-secure, 1= secure, non-secure callable */
#define SAU_INIT_NSC1       0
```

Σχήμα 2.3: Παράδειγμα κώδικα SAU

Εάν οποιαδήποτε περιοχή της μνήμης οριστεί ως «ασφαλής» είτε από την IDAU είτε από τη SAU, θα αντιμετωπίζεται ως ασφαλής. Όταν κάποιος ζητήσει πρόσβαση σε μια διεύθυνση από τον επεξεργαστή αυτή θα περάσει παράλληλα από τη SAU και την IDAU, για να ελεγχθεί εάν η πρόσβαση είναι "ασφαλής" ή "μη ασφαλής" πρόσβαση. Σε περίπτωση που το λογισμικό που εκτελείται με μη ασφαλή δικαιώματα πρόσβασης προσπαθήσει να αποκτήσει πρόσβαση σε μια θέση μνήμης που ορίζεται ως «ασφαλής» από τη SAU ή από την IDAU, δημιουργείται exception.[24]

2.4.5 Ασφαλής εκκίνηση σε μικροελεγκτές STM32

Η STMicroelectronics παρέχει για τις περισσότερες σειρές μικροελεγκτών της το X-CUBE-SBSFU, ένα πακέτο επέκτασης για SBSFU ή Secure Boot και Secure Firmware Update, το οποίο επιτρέπει την ασφαλή εκκίνηση του μικροελεγκτή STM32 και την ενημέρωση του ενσωματωμένου λογισμικού με νέες εκδόσεις, προσθέτοντας νέες δυνατότητες και διορθώνοντας πιθανά προβλήματα. Η διαδικασία εκκίνησης και ενημέρωσης εκτελείται με ασφαλή τρόπο για την αποτροπή μη εξουσιοδοτημένων ενημερώσεων και πρόσβασης σε εμπιστευτικά δεδομένα της συσκευής.

Η Secure Boot (υπηρεσίες Root of Trust) είναι ένας αμετάβλητος κώδικας, που εκτελείται πάντα μετά από επαναφορά συστήματος, που ελέγχει τις στατικές προ-στασίες STM32, ενεργοποιεί τις run-time προστασίες STM32 και στη συνέχεια επαληθεύει την αυθεντικότητα και την ακεραιότητα του κώδικα εφαρμογής πριν από κάθε εκτέλεση, προκειμένου να βεβαιωθεί ότι δεν είναι δυνατή η εκτέλεση μη έγκυρου ή κακόβουλου κώδικα.

Η εφαρμογή Secure Firmware Update λαμβάνει την εικόνα λογισμικού μέσω μιας διεπαφής UART με το πρωτόκολλο Ymodem, ελέγχει την αυθεντικότητά του και ελέγχει την ακεραιότητα του κώδικα πριν τον εγκαταστήσει. Παρέχει και τη δυνατότητα να ενεργοποιηθεί η δυνατότητα ενημέρωσης υλικολογισμικού over-the-air που χρησιμοποιείται συνήθως σε συσκευές IoT. Τα παραδείγματα που παρέχονται από τη ST μπορούν να διαμορφωθούν ώστε να χρησιμοποιούν ασύμμετρα ή συμμετρικά κρυπτογραφικά σχήματα με ή χωρίς κρυπτογράφηση υλικολογισμικού.[25]

Θα πρέπει να σημειωθεί ότι οι σειρές STM32 ικανές να εφαρμόσουν την ασφαλή εκκίνηση και την ασφαλή ενημέρωση υλικολογισμικού (SBSFU), με τη χρήση του πακέτου X-CUBE-SBSFU, είναι οι G0,L0,L1,G4,L4,L4+,F4,H7,F7 ενώ οι MCU της σειράς L5 πρέπει να έχουν εγκαταστημένο το TrustFirmware-M (TF-M) για να επιτευχθεί το ίδιο.

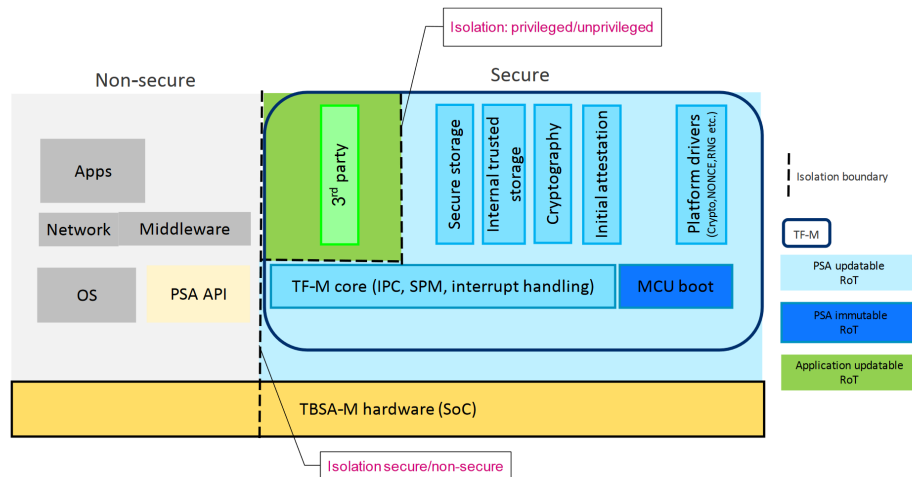
2.5 Trusted Firmware - M

Το TF-M είναι ένα λογισμικό ανοιχτού κώδικα που υλοποιεί το Secure Processing Environment για, μεταξύ άλλων, αρχιτεκτονικές Cortex-M33 και ευθυγραμμίζεται με τις οδηγίες για πιστοποίηση PSA. Το TF-M περιέχει πολλές ζώνες λογισμικού/υλικού που είναι απομονωμένες μεταξύ τους. Κάθε ζώνη έχει τον δικό της σκοπό και συνεργάζεται με άλλους για να πετύχει τον στόχο της ύπαρξης ενός αξιόπιστου υλικολογισμικού.

- PSA immutable RoT (Root of Trust): αμετάβλητη “Secure Boot and Secure Firmware Update” εφαρμογή που εκτελείται μετά από κάθε επαναφορά. Αυτή η εφαρμογή βασίζεται σε λογισμικό ανοιχτού κώδικα MCUBoot.
- PSA updatable RoT: “secure” εφαρμογή που υλοποιεί ένα σύνολο ασφαλών υπηρεσιών απομονωμένων στο ασφαλές περιβάλλον που μπορεί να κληθεί από τη μη ασφαλή εφαρμογή σε μη ασφαλή χρόνο εκτέλεσης της εφαρμογής με τη χρήση των PSA APIs:
 - Secure storage service
 - Internal trusted storage service
 - Cryptography service

– Initial attestation service

- Application updatable RoT: ασφαλείς υπηρεσίες τρίτων που είναι απομονωμένες σε ασφαλές/μη προνομιούχο περιβάλλον και οι οποίες μπορούν να κληθούν από τη μη ασφαλή εφαρμογή σε μη ασφαλή χρόνο εκτέλεσης της εφαρμογής.



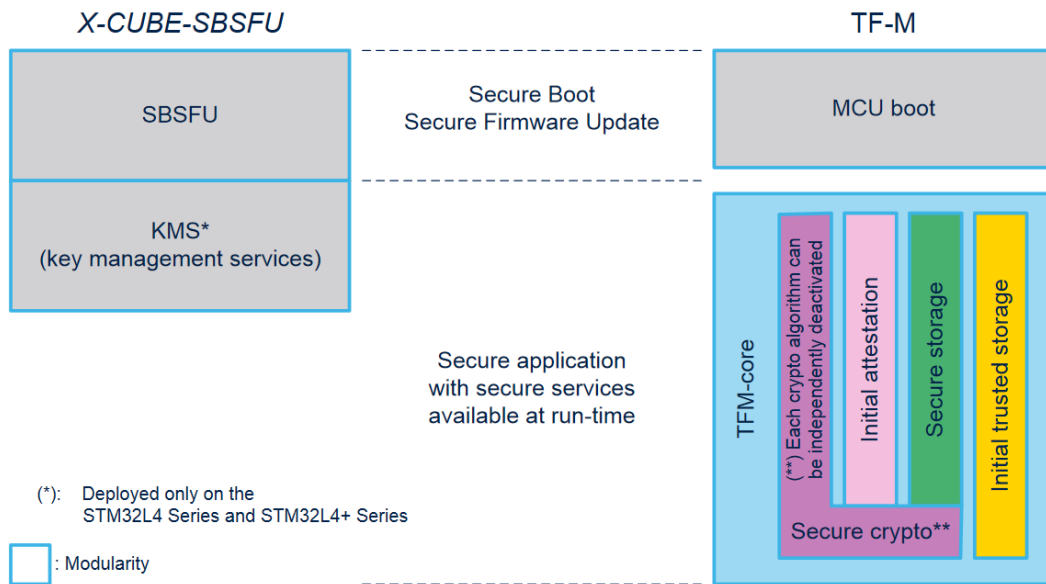
Σχήμα 2.4: Επισκόπηση TF-M

2.5.1 X-CUBE-SBSFU vs. TF-M

Το X-CUBE-SBSFU παρέχει μια υλοποίηση από την STMicroelectronics της Ασφαλούς εκκίνησης και της Ενημέρωσης Ασφαλούς υλικολογισμικού (SBSFU) και προαιρετικά μόνο για ορισμένες σειρές STM32, ασφαλή υπηρεσία KMS (υπηρεσίες διαχείρισης κλειδιών) που είναι διαθέσιμη κατά την εκτέλεση της εφαρμογής.

Από την άλλη η υλοποίηση TF-M παρέχει υπηρεσίες Ασφαλούς εκκίνησης και Ασφαλούς ενημέρωσης υλικολογισμικού που βασίζονται στη βιβλιοθήκη MCUBoot που είναι ανοιχτού κώδικα και προσφέρει ένα σύνολο ασφαλών υπηρεσιών κατά την εκτέλεση της εφαρμογής.

Το MCUBoot του TF-M μπορεί να συγκριθεί με το X-CUBE-SBSFU (χωρίς KMS) διότι προσφέρει παρόμοιες υπηρεσίες. Το X-CUBE-SBSFU KMS υποστηρίζει παρόμοιες υπηρεσίες με τις ασφαλείς υπηρεσίες κρυπτογράφησης TF-M, αλλά οι λίστες κρυπτογραφικών αλγορίθμων ή χαρακτηριστικών δεν είναι οι ίδιες και τα API διαφέρουν.



Σχήμα 2.5: X-CUBE-SBSFU vs. TF-M

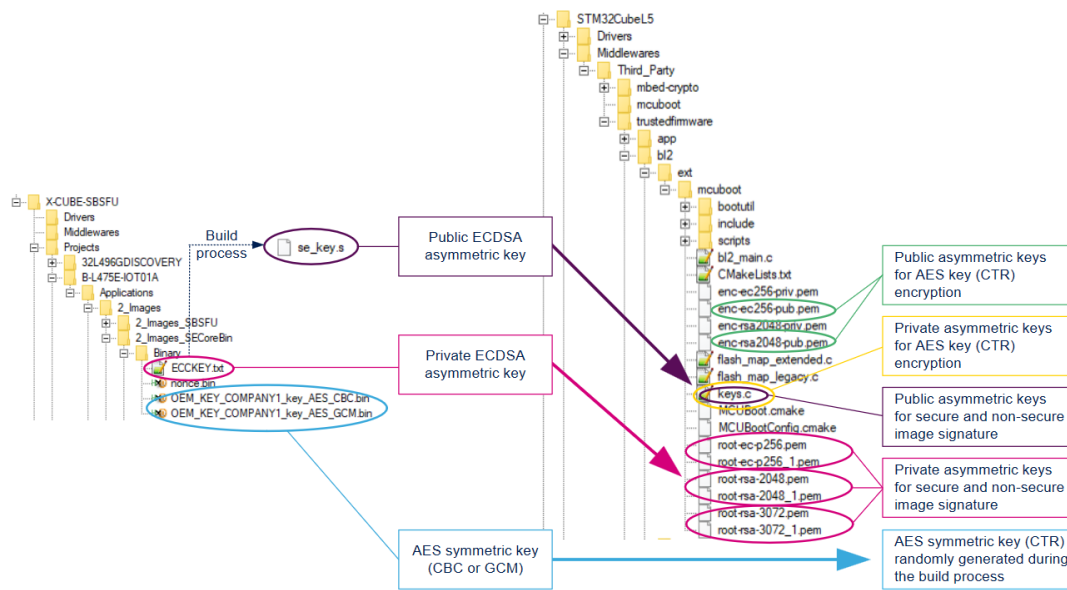
X-CUBE-SBSFU	TF-M
1 ή 2 υποδοχές ανά εικόνα. Νέα εικόνα μέσω τοπικού loader ή USER APP. Κρυπτογραφημένη εκτέλεση εικόνας σε εξωτερική μνήμη Flash.	1 ή 2 υποδοχές ανά εικόνα. Νέα εικόνα μέσω τοπικού loader ή USER APP. Κρυπτογραφημένη εκτέλεση εικόνας σε εξωτερική μνήμη Flash.
Ενιαία εικόνα υλικολογισμικού. Πλήρης ή μερική ενημέρωση.	Μία εικόνα υλικολογισμικού ή πολλές (2) εικόνες υλικολογισμικού (ασφαλείς και μη). Πλήρης ενημέρωση μόνο.
Συμμετρικό σχήμα κρυπτογράφησης. Ασύμμετρο σχήμα κρυπτογράφησης (ECDSA) ή συμμετρικό σχήμα κρυπτογράφησης, με ή χωρίς κρυπτογράφηση υλικολογισμικού.	Ασύμμετρο σχήμα κρυπτογράφησης (RSA ή ECDSA) με ή χωρίς κρυπτογράφηση υλικολογισμικού.

X-CUBE-SBSFU	TF-M
<p>Secure services</p> <ul style="list-style-type: none"> • 1 επίπεδο απομόνωσης • Διαχείριση μη ασφαλούς interrupt (μόνο σειρά STM32L4+) • Κύριες υπηρεσίες κρυπτογράφησης (μόνο σειρές STM32L4 και σειρές STM32L4+) 	<p>Secure services</p> <ul style="list-style-type: none"> • 2 επίπεδα απομόνωσης • Διαχείριση μη ασφαλούς interrupt • Ολοκληρωμένες υπηρεσίες κρυπτογράφησης (μόνο SW ή SW&HW) • Initial attestation • Secure Storage (κρυπτογράφηση/ακεραιότητα δεδομένων) • Internal trusted storage (ακεραιότητα δεδομένων) • Αρχιτεκτονικά έτοιμη να ενσωματώσει unprivileged υπηρεσίες εφαρμογών

Πίνακας 2.4: X-CUBE-SBSFU vs. TF-M top-level features

Στο TF-M SBSFU στη βιβλιοθήκη STM32CubeL5 έκδοση 1.4.0, για την υπογραφή εικόνας υλικολογισμικού, υπάρχουν δύο ασύμμετρα κλειδιά RSA ή ECDSA (ένα για ασφαλή εικόνα και ένα για μη ασφαλή εικόνα), σε σύγκριση με ένα ασύμμετρο κλειδί ECDSA στο X-CUBE-SBSFU. Πρέπει να σημειωθεί ότι σε αντίθεση με το X-CUBE-SBSFU, τα δημόσια ασύμμετρα κλειδιά δε δημιουργούνται αυτόματα κατά τη διαδικασία κατασκευής του STM32CubeL5 SBSFU, αλλά πρέπει να παρέχονται από τον χρήστη μαζί με τα ιδιωτικά ασύμμετρα κλειδιά. Το TF-M SBSFU στη STM32CubeL5 V1.4.0 υποστηρίζει κρυπτογράφηση υλικολογισμικού με κρυπτογράφηση AES-CTR. Σε σύγκριση με το X-CUBE-SBSFU, το κλειδί AES-CTR δεν υπάρχει στα εξατομικευμένα δεδομένα, αλλά δημιουργείται τυχαία κατά τη διάρκεια κάθε διαδικασίας κατασκευής, είναι κρυπτογραφημένο (RSA-OAEP ή ECIES-P256) και παρέχεται κατευθείαν στην ίδια την εικόνα υλικολογισμικού.

Εκτός από τα κλειδιά ελέγχου ταυτότητας εικόνας υλικολογισμικού, τα πρόσθετα δεδομένα απαιτούν εξατομίκευση για την εφαρμογή TFM: κλειδί EAT, HUK και Instance ID. Αυτά τα δεδομένα απαιτούνται για την υπηρεσία initial attestation του TF-M. Αυτά τα δεδομένα, μαζί με τα ασύμμετρα κλειδιά για την υπογραφή εικόνων και την αποκρυπτογράφηση κλειδιού AES CTR, ομαδοποιούνται στην αμετάβλητη περιοχή Flash (περιοχή δεδομένων εξατομίκευσης), η οποία πρέπει να εξατομικεύεται για κάθε συσκευή στην παραγωγή, πριν την ενεργοποίηση της τελικής διαμόρφωσης ασφαλείας.[26]



Σχήμα 2.6: Firmware image keys personalization

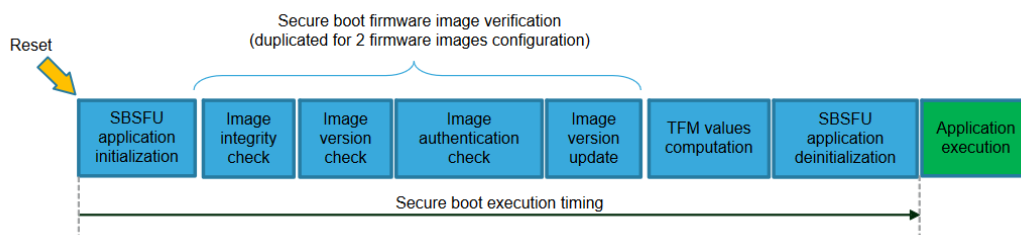
2.5.2 SBSFU με το TF-M

HUK Το κλειδί Huk (Μοναδικό κλειδί ιεραρχίας) είναι ένα μυστικό κρυπτογραφικό κλειδί που χρησιμοποιείται για την ασφάλεια των δεδομένων σε έναν συγκεκριμένο τομέα ασφαλείας. Το κλειδί Huk προέρχεται συνήθως από ένα RoT και είναι μοναδικό για τον τομέα, για τον οποίο δημιουργήθηκε. Χρησιμοποιείται για την προστασία ευαίσθητων δεδομένων εντός του τομέα και δεν κοινοποιείται με κανέναν άλλο τομέα. Στο TF-M το HUK κρυπτογραφεί τα δεδομένα του bootloader που χρειάζεται να μοιραστούν με τα secure services στην ασφαλή περιοχή.

Secure Boot

Ο bootloader ξεκινά όταν η CPU αποδεσμευτεί από την επαναφορά. Λειτουργεί σε ασφαλή λειτουργία. Πραγματοποιεί έλεγχο ταυτότητας της εικόνας υλικολογισμικού με επικύρωση κατακερματισμού (SHA-256) και ψηφιακής υπογραφής (RSA-2048). Τα μεταδεδομένα της εικόνας παραδίδονται μαζί με την ίδια την εικόνα σε μια ενότητα κεφαλίδας και τρέιλερ. Σε περίπτωση επιτυχούς ελέγχου ταυτότητας, ο bootloader περνά την εκτέλεση στην ασφαλή εικόνα. Η εκτέλεση δεν επιστρέφει ποτέ στον bootloader μέχρι την επόμενη επαναφορά.

Ο bootloader μπορεί να χειριστεί τις ασφαλείς και μη ασφαλείς εικόνες ανεξάρτητα (εκκίνηση πολλαπλών εικόνων) ή μαζί (εκκίνηση μίας εικόνας). Σε περίπτωση εκκίνησης πολλαπλών εικόνων υπογράφονται ανεξάρτητα με διαφορετικά κλειδιά και μπορούν να ενημερωθούν ξεχωριστά. Σε περίπτωση εκκίνησης μεμονωμένης εικόνας, η ασφαλής και μη ασφαλής εικόνα αντιμετωπίζεται ως ένα ενιαίο δυαδικό, επομένως πρέπει να είναι συνεχόμενα στη μνήμη της συσκευής. Σε αυτήν την περίπτωση υπογράφονται μαζί και επίσης μπορούν να ενημερωθούν μόνο μαζί. Για να υπάρχουν τα ίδια artefacts στο τέλος της κατασκευής ανεξάρτητα από τον τρόπο χειρισμού των εικόνων (ανεξάρτητα ή μαζί), οι εικόνες είναι πάντα συνενωμένες. Σε περίπτωση εκκίνησης μίας εικόνας, πρώτα ενώνονται και μετά υπογράφονται. Σε περίπτωση εκκίνησης πολλαπλών εικόνων, πρώτα υπογράφονται ξεχωριστά και μετά συνδέονται.[27]



Σχήμα 2.7: Secure Boot χρονοδιάγραμμα εκτέλεσης

Image versioning Το όρισμα αριθμός έκδοσης εικόνας βρίσκεται στην κεφαλίδα της υπογεγραμμένης εικόνας και είναι προαιρετικό. Αν παραληφθεί, τότε οι αριθμοί έκδοσης των εικόνων που δημιουργούνται στον ίδιο κατάλογο θα αλλάξουν αυτόματα. Σε αυτήν την περίπτωση, το τελευταίο στοιχείο (ο αριθμός έκδοσης) αυξάνεται αυτόματα από το προηγούμενο: 0,0,0+1 -> 0,0,0+2, για όσες φορές η

έκδοση ξανά τρέξει, μέχρι να δοθεί ρητά ένας αριθμός. Εάν υπάρχει αυτόματη έκδοση έκδοσης και στη συνέχεια δοθεί αριθμός έκδοσης εικόνας για πρώτη φορά, ο νέος αριθμός θα έχει προτεραιότητα και θα χρησιμοποιείται αντ' αυτού. Όλες οι επόμενες εκδόσεις εικόνας ορίζονται στη συνέχεια με βάση αυτόν τον αριθμό και ο αριθμός θα σταματήσει να αυξάνεται. Για να ενεργοποίηση ξανά της αυτόματης έκδοσης εικόνας, χρειάζεται μια καθαρή έκδοση χωρίς ορισμένο τον αριθμό έκδοσης εικόνας.

Security counter Κάθε υπογεγραμμένη εικόνα περιέχει επίσης έναν μετρητή ασφαλείας στα μεταδεδομένα της. Χρησιμοποιείται από το bootloader και στόχος του είναι να έχει έναν ανεξάρτητο (από την έκδοση εικόνας) μετρητή για να διασφαλίζει την προστασία επαναφοράς συγκρίνοντας τον μετρητή ασφαλείας της νέας εικόνας με τον μετρητή ασφαλείας της αρχικής (επί του παρόντος ενεργού) εικόνας κατά τη διαδικασία αναβάθμισης εικόνας. Προστίθεται στο μανιφέστο (στην περιοχή TLV που προσαρτάται στο τέλος της εικόνας) από ένα από τα σενάρια Python κατά την υπογραφή της εικόνας. Η τιμή του μετρητή ασφαλείας είναι ζωτικής σημασίας για την ασφάλεια και βρίσκεται στο προστατευμένο τμήμα της εικόνας.

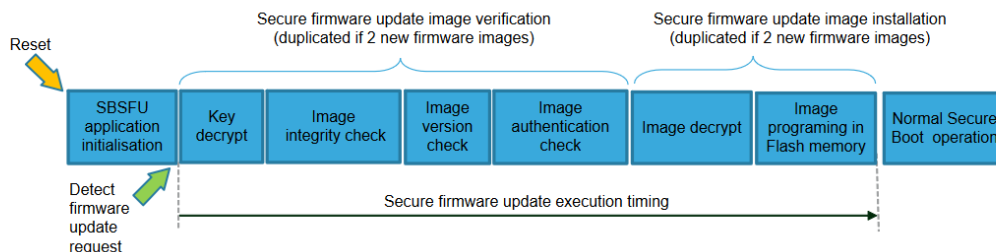
NV Counters NV Counters σύμφωνα με το Trusted Base System Architecture (TBSA-M) είναι ένας μετρητής με τις ακόλουθες ιδιότητες:

- Πρέπει να είναι δυνατή η αύξηση ενός μετρητή μόνο μέσω μιας αξιόπιστης πρόσβασης.
- Πρέπει να είναι δυνατή μόνο η αύξηση ενός μετρητή. Δεν πρέπει να είναι δυνατή η μείωση του.
- Όταν ένας μετρητής φτάσει στη μέγιστη τιμή του, δεν πρέπει να ξεκινάει από την αρχή και δεν πρέπει να είναι δυνατές περαιτέρω αλλαγές.
- Ένας μετρητής πρέπει να είναι non-volatile και η αποθηκευμένη τιμή πρέπει να επιβιώνει όταν το board χάνει το ρεύμα κατά όλη τη διάρκεια ζωής της συσκευής.

Έμπιστοι non-volatile counters (NV Counters) μπορούν να χρησιμοποιηθούν για την αποθήκευση της τιμής των μετρητών ασφαλείας ανά εικόνα λογισμικού με δυνατότητα ενημέρωσης τους σε περίπτωση αναβάθμισης. Στην ιδανική περίπτωση, όλες οι

εικόνες λογισμικού που ενημερώνονται θα πρέπει να διαθέτουν ξεχωριστό μετρητή ασφαλείας.

Secure Firmware Update



Σχήμα 2.8: Secure Firmware Update χρονοδιάγραμμα εκτέλεσης

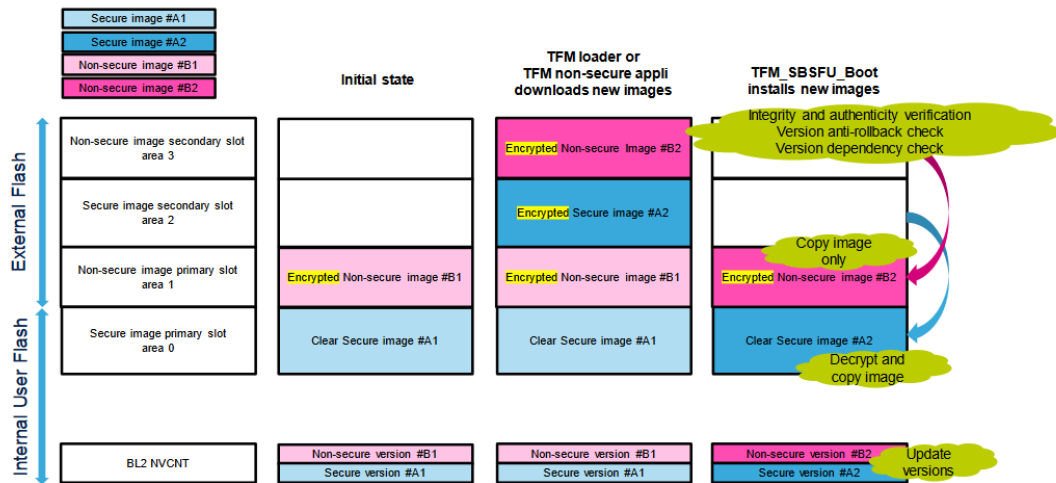
Το MCUBoot χειρίζεται μόνο τον έλεγχο γνησιότητας υλικολογισμικού μετά την εκκίνηση και την ανταλλαγή εικόνων κατά της διαδικασίας ενημέρωσης υλικολογισμικού. Η λήψη της νέας έκδοσης του υλικολογισμικού είναι εκτός πεδίου εφαρμογής για το MCUBoot. Το MCUBoot υποστηρίζει τρεις διαφορετικούς τρόπους μετάβασης στο νέο υλικολογισμικό και θεωρείται ότι οι εικόνες υλικολογισμικού εκτελούνται επιτόπου (XIP). Η προεπιλεγμένη συμπεριφορά είναι η αναβάθμιση εικόνας που βασίζεται σε αντικατάσταση. Σε αυτήν την περίπτωση, το ενεργό υλικολογισμικό εκτελείται πάντα από την κύρια υποδοχή και η δευτερεύουσα υποδοχή είναι μια περιοχή σταδιοποίησης για νέες εικόνες. Πριν από την εκτέλεση της νέας εικόνας υλικολογισμικού, το περιεχόμενο της κύριας υποδοχής πρέπει να αντικατασταθεί με το περιεχόμενο της δευτερεύουσας υποδοχής (η νέα εικόνα υλικολογισμικού). Η δεύτερη επιλογή είναι η στρατηγική εναλλαγής εικόνας όταν το περιεχόμενο των δύο υποδοχών μνήμης πρέπει να αντικατασταθεί φυσικά. Αυτή η επιλογή απαιτεί να οριστεί η περιοχή scratch στη διάταξη μνήμης. Η τρίτη επιλογή είναι η έκδοση άμεσης εκτέλεσης επί τόπου, η οποία εξαλείφει την πολυπλοκότητα της εναλλαγής εικόνας και της διαχείρισής της. Η ενεργή εικόνα μπορεί να εκτελεστεί από οποιαδήποτε υποδοχή μνήμης, αλλά το νέο υλικολογισμικό πρέπει να συνδεθεί με τον χώρο διευθύνσεων της κατάλληλης (επί του παρόντος ανενεργής) υποδοχής μνήμης.[27]

1. Overwrite operation: Η ενεργή εικόνα αποθηκεύεται στην κύρια υποδοχή και αυτή η εικόνα ξεκινά πάντα από τον bootloader. Επομένως, οι εικόνες πρέπει να συνδέονται με την κύρια υποδοχή. Εάν ο bootloader βρει μια έγκυρη εικόνα

στη δευτερεύουσα υποδοχή, η οποία έχει επισημανθεί για αναβάθμιση, τότε το περιεχόμενο της κύριας υποδοχής θα αντικατασταθεί απλώς με το περιεχόμενο της δευτερεύουσας υποδοχής, πριν ξεκινήσει η νέα εικόνα από την κύρια υποδοχή. Μετά την επιτυχή αντικατάσταση του περιεχομένου της κύριας υποδοχής, η κεφαλίδα και το τρέιλερ της νέας εικόνας στη δευτερεύουσα υποδοχή διαγράφονται για να αποτραπεί η ενεργοποίηση μιας άλλης περιττής αναβάθμισης εικόνας μετά από επανεκκίνηση. Η λειτουργία αντικατάστασης είναι ασφαλής έναντι αποτυχίας και ανθεκτική σε αστοχίες διακοπής ρεύματος.

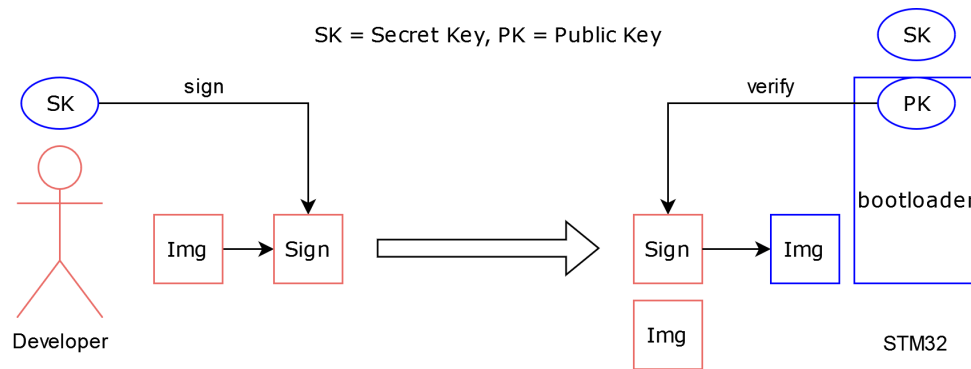
2. Swapping operation: Με τη στρατηγική αναβάθμισης εναλλαγής εικόνας, η ενεργή εικόνα αποθηκεύεται επίσης στην κύρια υποδοχή και θα ξεκινά πάντα από τον bootloader. Εάν ο bootloader βρει μια έγκυρη εικόνα στη δευτερεύουσα υποδοχή, η οποία έχει επισημανθεί για αναβάθμιση, τότε τα περιεχόμενα της κύριας υποδοχής και της δευτερεύουσας υποδοχής θα εναλλάσσονται, πριν ξεκινήσει η νέα εικόνα από την κύρια υποδοχή. Η περιοχή Scratch χρησιμοποιείται ως χώρος προσωρινής αποθήκευσης κατά την εναλλαγή εικόνων. Το σήμα ενημέρωσης από τη δευτερεύουσα υποδοχή αφαιρείται όταν η εναλλαγή είναι επιτυχής. Ο bootloader μπορεί να επαναφέρει την εναλλαγή ως εναλλακτικό μηχανισμό για την ανάκτηση της προηγούμενης λειτουργικής έκδοσης υλικολογισμικού μετά από μια ελαττωματική ενημέρωση.
3. Direct execute-in-place operation: Στη λειτουργία direct-xip, η ενεργή σημαία εικόνας μετακινείται μεταξύ των υποδοχών κατά την αναβάθμιση υλικολογισμικού. Ο loader αναβάθμισης υλικολογισμικού, ο οποίος κατεβάζει τη νέα εικόνα, πρέπει να γνωρίζει ποια υποδοχή φιλοξενεί το ενεργό υλικολογισμικό και ποια λειτουργεί ως περιοχή σταδίου και είναι υπεύθυνος για τη λήψη της εικόνας στην κατάλληλη θέση. Κατά την εκκίνηση, το MCUBoot επιθεωρεί τον αριθμό έκδοσης στην κεφαλίδα της εικόνας και μεταβιβάζει την εκτέλεση στη νεότερη έκδοση υλικολογισμικού. Η νέα εικόνα πρέπει να επισημανθεί για αναβάθμιση, το οποίο γίνεται αυτόματα από τα Python scripts τη στιγμή της μεταγλώττισης.

Η επαλήθευση εικόνας γίνεται με τον ίδιο τρόπο σε όλους τους τρόπους λειτουργίας. Εάν η νέα εικόνα αποτύχει κατά τον έλεγχο ταυτότητας, το MCUBoot διαγράφει την υποδοχή μνήμης και ξεκινά την άλλη εικόνα, μετά από επιτυχή έλεγχο ταυτότητας.

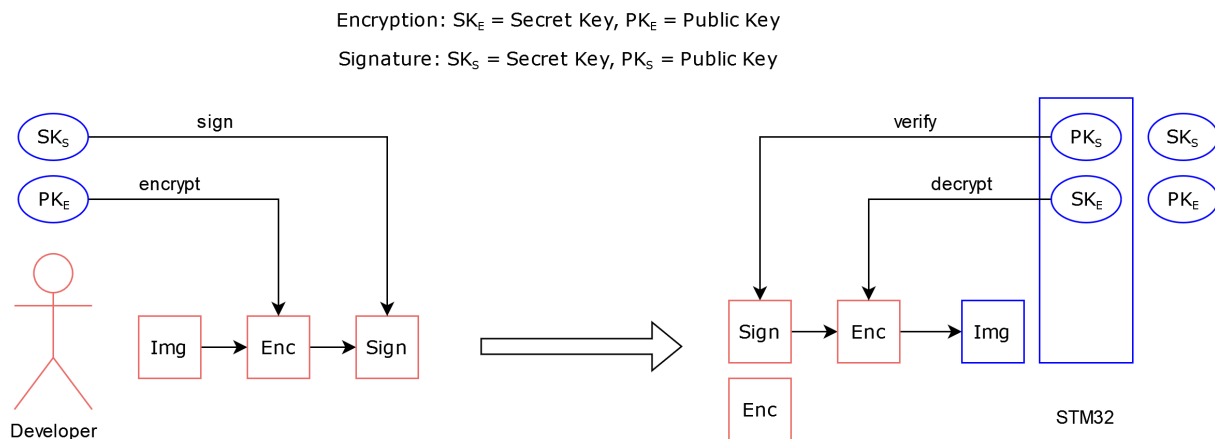


Σχήμα 2.9: Διαδικασία λήψης και εγκατάστασης νέου υλικολογισμικού (Overwrite operation)

Υπογραφή εικόνας Το MCUBoot διασφαλίζει την ακεραιότητα και την αυθεντικότητα των εικόνων υλικολογισμικού πριν από την εκκίνηση τους. Χρησιμοποιεί ένα Python script που ονομάζεται `imgtool.py` για την υπογραφή εικόνων με ένα ιδιωτικό κλειδί, το οποίο θα πρέπει να διατηρείται μυστικό και να μην ενσωματώνεται στο bootloader ή τις εικόνες του υλικολογισμικού. Το δημόσιο κλειδί είναι ενσωματωμένο στο bootloader και χρησιμοποιείται για την επαλήθευση των υπογραφών. Το script υποστηρίζει διαφορετικούς τύπους κλειδιών όπως RSA και ECDSA και προσθέτει μεταδεδομένα στην κεφαλίδα της εικόνας, συμπεριλαμβανομένης της έκδοσης, του μεγέθους και του hash της εικόνας. Η κεφαλίδα περιέχει επίσης έναν μαγικό αριθμό που προσδιορίζει την εικόνα ως έγκυρη εικόνα MCUBoot. Το τρέιλερ, το οποίο δεν αποτελεί μέρος του αρχείου εικόνας, είναι γραμμένο από τον bootloader και περιέχει σημαίες και μετρητές που χρησιμοποιούνται για τον έλεγχο της διαδικασίας ενημέρωσης υλικολογισμικού, καθώς και έναν μαγικό αριθμό που υποδεικνύει την παρουσία μιας νέας εικόνας στη δευτερεύουσα υποδοχή. Με αυτές τις δυνατότητες, το MCUBoot παρέχει έναν ασφαλή και αξιόπιστο τρόπο ενημέρωσης εικόνων υλικολογισμικού σε συσκευές.



Σχήμα 2.10: Απεικόνιση διαδικασίας υπογραφής εικόνας



Σχήμα 2.11: Απεικόνιση διαδικασίας κρυπτογράφησης και υπογραφής εικόνας

2.5.3 Ασφαλείς υπηρεσίες στο TF-M

Οι ασφαλείς υπηρεσίες είναι ένα σύνολο υπηρεσιών, που μπορούν να κληθούν σε μη ασφαλή χρόνο εκτέλεσης και διαχειρίζονται κρίσιμα στοιχεία που είναι απομονωμένα από τη μη ασφαλή εφαρμογή. Η μη ασφαλής εφαρμογή δεν μπορεί να έχει απευθείας πρόσβαση σε κανένα από τα κρίσιμα στοιχεία, αλλά μπορεί να χρησιμοποιήσει μόνο ασφαλείς υπηρεσίες που χρησιμοποιούν τα κρίσιμα στοιχεία. Ασφαλείς υπηρεσίες παρέχονται με δύο επίπεδα απομόνωσης χάρη στη χρήση privileged/unprivileged τρόπου λειτουργίας. (ο επεξεργαστής μπορεί να περιορίσει ή να αποκλείσει την πρόσβαση σε ορισμένους πόρους εκτελώντας κώδικα σε privileged ή unprivileged λειτουργία)

Secure storage service (SST)

Η υπηρεσία ασφαλούς αποθήκευσης TF-M (SST) εφαρμόζει PSA protected storage APIs. Η υπηρεσία υποστηρίζεται από απομόνωση υλικού της πρόσβασης flash και,

στην τρέχουσα έκδοση, βασίζεται σε υλικό για την απομόνωση της περιοχής φλας από μη ασφαλή πρόσβαση. Η υπηρεσία SST παρέχει ένα μη ιεραρχικό μοντέλο αποθήκευσης, ως σύστημα αρχείων, όπου όλα τα στοιχεία διαχειρίζονται από μια λίστα μεταδεδομένων.

Internal trusted storage service (ITS)

Η υπηρεσία εσωτερικής αξιόπιστης αποθήκευσης (ITS) TF-M εφαρμόζει PSA internal trusted storage APIs που επιτρέπουν την εγγραφή δεδομένων σε μια ενσωματωμένη περιοχή μνήμης Flash που θα απομονώνεται από μη ασφαλείς εφαρμογές μέσω των μηχανισμών προστασίας της ασφάλειας υλικού.

Secure cryptographic service

Η υπηρεσία TF-M crypto παρέχει μια υλοποίηση του PSA crypto API βασισμένη στη βιβλιοθήκη mbed-crypto. Περισσότερες πληροφορίες για το PSA crypto API ή την mbed-crypto βρίσκονται απευθείας στο MbedCrypto GitHub repo.

Initial attestation service

Η υπηρεσία initial attestation επιτρέπει στην εφαρμογή να αποδείξει την ταυτότητα της συσκευής κατά τη διάρκεια μιας διαδικασίας ελέγχου ταυτότητας. Η υπηρεσία μπορεί να δημιουργήσει ένα διακριτικό βεβαίωσης (entity attestation token, EAT) κατόπιν αιτήματος, το οποίο περιέχει ένα προκαθορισμένο σύνολο από δεδομένα μοναδικά για τη συσκευή (Claims). Στην ασύμμετρη λειτουργία η συσκευή πρέπει να περιέχει ένα ζεύγος κλειδιών, το οποίο είναι μοναδικό ανά συσκευή. Το διακριτικό βεβαίωσης κωδικοποιείται με τον αλγόριθμο CBOR και υπογράφεται με το ιδιωτικό μέρος του ζεύγους κλειδιών χρησιμοποιώντας τον αλγόριθμο COSE. Το δημόσιο κλειδί χρησιμοποιείται για την επαλήθευση της αυθεντικότητας του διακριτικού. Τα στοιχεία στο διακριτικό χρησιμοποιούνται για την επαλήθευση της ακεραιότητας της συσκευής και την αξιολόγηση της αξιοπιστίας της. Η παροχή των κλειδιών λαμβάνει μέρος κατά την κατασκευή του προϊόντος.[28]

Claims Το διακριτικό βεβαίωσης αποτελείται από claims. Το ακόλουθο σταθερό σύνολο από claims περιλαμβάνονται στο token:

- Auth challenge: Το challenge που στέλνει ο χρήστης. Σκοπός του είναι να αλλάξει την απάντηση του initial attestation καθώς όλα τα άλλα claims παραμένουν σταθερά. Μπορεί να είναι ένα μόνο nonce ή ένας κατακερματισμός ενός nonce και πιστοποιημένα δεδομένα.
- Instance ID: Ένα μοναδικό αναγνωριστικό που αντιπροσωπεύει τη συσκευή. Είναι το hash του δημόσιου κλειδιού σε ασύμμετρη και το hash του συμμετρικού κλειδιού σε συμμετρική λειτουργία.
- Verification service indicator: Ένα προαιρετικό claim που μπορεί να χρησιμοποιηθεί από ένα Τρίτο μέρος για να εντοπίσει μια υπηρεσία επικύρωσης για το διακριτικό. Είναι μια συμβολοσειρά κειμένου ή διεύθυνση URL.
- Profile definition: Ένα προαιρετικό claim που περιέχει το όνομα ενός εγγράφου που περιγράφει το «προφίλ» του διακριτικού, συμπεριλαμβανομένων των claims, της χρήσης, της επαλήθευσης και της υπογραφής του διακριτικού.
- Implementation ID: Ένα claim που προσδιορίζει μοναδικά το PSA RoT και μπορεί να χρησιμοποιηθεί από μια υπηρεσία επαλήθευσης για τον εντοπισμό λεπτομερειών της διαδικασίας επαλήθευσης.
- Client ID: Ένα claim που αντιπροσωπεύει ποιος κάλεσε το initial attestation API (πχ non-secure νήμα).
- Security lifecycle: Αντιπροσωπεύει την τρέχουσα κατάσταση του κύκλου ζωής της συσκευής.
- Hardware version: Παγκοσμίως μοναδικός αριθμός σε μορφή EAN-13 που προσδιορίζει το GDSII που πήγε στην κατασκευή, HW και ROM.
- Boot seed: Ένα claim που αντιπροσωπεύει μια τυχαία τιμή που δημιουργήθηκε κατά την εκκίνηση του συστήματος για να διαφοροποιήσει τις αναφορές από διαφορετικές περιόδους λειτουργίας συστήματος.
- Software components: Ένα προαιρετικό claim που αντιπροσωπεύει την κατάσταση λογισμικού του συστήματος
- No software measurements: Ένας προαιρετικό claim που υποδεικνύει τη σκόπιμη απουσία μετρήσεων λογισμικού, με τιμή 1. Ή.

- **Measurements:** Με κάθε μέτρηση να χρειάζεται τύπο, τιμή και περιγραφή.

COSE header Μια κεφαλίδα COSE είναι ένα σύνολο παραμέτρων που παρέχουν πρόσθετες πληροφορίες σχετικά με ένα κωδικοποιημένο μήνυμα ή δομή δεδομένων CBOR που έχει υπογραφεί ή κρυπτογραφηθεί χρησιμοποιώντας COSE. Η κεφαλίδα περιέχει μεταδεδομένα σχετικά με τις ιδιότητες ασφαλείας του μηνύματος, όπως τον αλγόριθμο που χρησιμοποιείται για την υπογραφή ή την κρυπτογράφηση, το κλειδί που χρησιμοποιείται για την υπογραφή ή την κρυπτογράφηση και οποιεσδήποτε άλλες παραμέτρους που απαιτούνται για τη σωστή ερμηνεία του μηνύματος.

2.5.4 Μέτρα προστασίας και στρατηγική ασφάλειας

Η κρυπτογραφία εξασφαλίζει ακεραιότητα, πιστοποίηση και εμπιστευτικότητα. Ωστόσο, η χρήση της κρυπτογραφίας από μόνη της δεν είναι αρκετή, απαιτείται ένα σύνολο μέτρων και στρατηγική σε επίπεδο συστήματος για την προστασία κρίσιμων λειτουργιών, ευαίσθητων δεδομένων (όπως ένα μυστικό κλειδί) και τη ροή εκτέλεσης, προκειμένου να αντισταθεί το σύστημα σε πιθανές επιθέσεις.

Το παράδειγμα STM32CubeL5 TFM χρησιμοποιεί μια στρατηγική ασφάλειας που βασίζεται στις ακόλουθες έννοιες:

- Μόνο ένα σημείο εκκίνησης σε κάθε επαναφορά: Η εκτέλεση του κώδικα ξεκινάει με τον κώδικα ασφαλούς εκκίνησης.
- Ο κώδικας TFM_SBSFU_Boot είναι αμετάβλητος: Δεν υπάρχει δυνατότητα τροποποίησης ή αλλαγής τους μόλις ενεργοποιηθεί πλήρως η ασφάλεια.
- 3 προστατευμένοι/απομονωμένοι τομείς:
 - Secure / privileged: Για την εκτέλεση του immutable RoT PSA. Αυτός ο τομέας κρύβεται μόλις ολοκληρωθεί η εκτέλεση του immutable PSA RoT.
 - Secure / privileged: Για την εκτέλεση του updatable RoT PSA.
 - Secure / unprivileged: Για την εκτέλεση του application updatable RoT.
- Περιορισμός της επιφάνειας εκτέλεσης σύμφωνα με την κατάσταση της εφαρμογής:
 - Από την επαναφορά μέχρι να επαληθευτεί η εγκατεστημένη εφαρμογή: επιτρέπεται μόνο η εκτέλεση κώδικα εκκίνησης TFM_SBSFU.

- Μόλις επαληθευτεί η εγκατάσταση της εφαρμογής: επιτρέπεται η εκτέλεση κώδικα εφαρμογής (ασφαλές μέρος και μη ασφαλές μέρος).
- Κατάργηση της πρόσβασης στο JTAG.

Προστασία από εξωτερικές επιθέσεις

Οι εξωτερικές επιθέσεις αναφέρονται σε επιθέσεις που προκαλούνται από εξωτερικά εργαλεία, όπως προγράμματα εντοπισμού σφαλμάτων ή ανιχνευτές, που προσπαθούν να αποκτήσουν πρόσβαση στη συσκευή. Τα μέτρα που προστατεύουν το προϊόν από εξωτερικές επιθέσεις είναι τα εξής:

- RDP (read data protection): Το επίπεδο RDP 1 χρησιμοποιείται για να διασφαλιστεί ότι το πρόγραμμα εντοπισμού σφαλμάτων JTAG δεν μπορεί να έχει πρόσβαση σε οποιοδήποτε ασφαλές ή προστατευμένο μέρος της συσκευής:
 - Απαγορεύεται ασφαλές JTAG debug.
 - Απαγορεύεται η πρόσβαση σε προστατευμένη μνήμη (Μνήμη flash, SRAM2 και εφεδρικοί καταχωρητές).
 - Το JTAG μπορεί να έχει πρόσβαση μόνο στο μη ασφαλές SRAM1 και σε όλους τους μη ασφαλείς περιφερειακούς καταχωρητές.
- Κλείδωμα εκκίνησης: Το option byte BOOT_LOCK χρησιμοποιείται για τον καθορισμό του σημείου εισόδου σε μια συγκεκριμένη θέση μνήμης και δεν αλλάζει αφού ρυθμιστεί.
- Προστατευμένη SRAM2: Η SRAM2 προστατεύεται αυτόματα από εισβολή όταν το σύστημα διαμορφωθεί στο επίπεδο RDP 1. Το περιεχόμενο της SRAM2 διαγράφεται μόλις εντοπιστεί μια εισβολή. Επιπλέον, το περιεχόμενο SRAM2 μπορεί να προστατεύεται από εγγραφές (το περιεχόμενο "κλειδώνει" αλλά μπορεί να διαβαστεί) μέχρι την επόμενη επαναφορά ενεργοποιώντας το bit κλειδώματος. Στο παράδειγμα TFM, το σύστημα έχει διαμορφωθεί ώστε να χρησιμοποιεί την προστατευμένη SRAM2 για κοινή χρήση πληροφοριών μεταξύ της εφαρμογής TFM_SBSFU_Boot και της ασφαλούς εφαρμογής.

Άλλα μέτρα που θα μπορούσαν να χρησιμοποιηθούν για την προστασία του συστήματος από εξωτερικές επιθέσεις, αλλά το τρέχον παράδειγμα TFM δεν τα χρησιμοποιεί:

- Anti-tamper protection: Η προστασία θα μπορούσε να χρησιμοποιηθεί για τον εντοπισμό φυσικών ενεργειών παραβίασης στη συσκευή και να λάβει τα σχετικά αντίμετρα. Σε περίπτωση εντοπισμού παραβίασης, το TFM_SBSFU_Boot θα μπορούσε να αναγκάσει το σύστημα σε επανεκκίνηση.
- DAP (debug access port) protection: Η προστασία debug είναι υπεύθυνη για την απενεργοποίηση του DAP (Debug Access Port). Μόλις απενεργοποιηθεί, τα JTAG pins δεν είναι πλέον συνδεδεμένα στον εσωτερικό bus του STM32. Το DAP απενεργοποιείται αυτόματα στο επίπεδο RDP 2.
- IWDG (independent watchdog): Ένας μετρητής που μετράει αντίστροφα και μόλις ξεκινήσει δεν μπορεί να σταματήσει. Πρέπει να ανανεώνεται περιοδικά αλλιώς προκαλεί επαναφορά του συστήματος.

Προστασία από εσωτερικές επιθέσεις

Οι εσωτερικές επιθέσεις αναφέρονται σε επιθέσεις που προκαλούνται από κώδικα που τρέχει στο STM32. Οι επιθέσεις μπορεί να οφείλονται είτε σε κακόβουλο υλικολογισμικό που εκμεταλλεύεται σφάλματα ή παραβιάσεις ασφάλειας, είτε σε ανεπιθύμητες λειτουργίες. Τα μέτρα που προστατεύουν το προϊόν από εσωτερικές επιθέσεις είναι τα εξής:

- TZ: Ο πυρήνας της CPU υποστηρίζει 2 τρόπους λειτουργίας (ασφαλή και μη). Όταν ο πυρήνας βρίσκεται σε μη ασφαλή λειτουργία, δεν μπορεί να έχει πρόσβαση σε πόρους SMT32L5 που έχουν ρυθμιστεί ως ασφαλείς.
- MPU: Η MPU είναι ένας μηχανισμός προστασίας μνήμης που επιτρέπει τον καθορισμό συγκεκριμένων δικαιωμάτων πρόσβασης για οποιονδήποτε πόρο αντιστοιχισμένο στη μνήμη της συσκευής: Μνήμη Flash, SRAM και περιφερειακούς καταχωρητές. Η Secure MPU χρησιμοποιείται για τον έλεγχο της πρόσβασης CPU σε ασφαλή λειτουργία και η non-secure-MPU χρησιμοποιείται για τον έλεγχο της πρόσβασης CPU σε μη ασφαλή λειτουργία.
- SAU: Η SAU είναι μια μονάδα υλικού συνδεδεμένη με τον πυρήνα (όπως η MPU), υπεύθυνη για τον καθορισμό του χαρακτηριστικού ασφαλείας της συναλλαγής AHB5 (προηγμένος δίαυλος υψηλής απόδοσης).

- GTZC: παρέχει μηχανισμούς ρύθμισης μνημών και περιφερειακών ώστε να είναι secure ή non-secure και privileged ή unprivileged.

Κατά την εκτέλεση του κώδικα TFM_SBSFU_Boot, μόνο η περιοχή στη flash που αντιστοιχεί στον κώδικα εκκίνησης TFM_SBSFU_μπορεί να εκτελεστεί από τη CPU σε ασφαλή λειτουργία, οι άλλες περιοχές μνήμης (μνήμη Flash και SRAM) έχουν μόνο δικαιώματα πρόσβασης για ανάγνωση/εγγραφή. Πριν από την εκκίνηση της επαληθευμένης εφαρμογής, η εφαρμογή TFM_SBSFU_Boot διαμορφώνει εκ νέου το σύστημα έτσι ώστε η επιφάνεια εκτέλεσης να επεκταθεί με την περιοχή στη flash που αντιστοιχεί στην επαληθευμένη εφαρμογή (τόσο ασφαλές μέρος όσο και μη ασφαλές μέρος), ενώ οι άλλες περιοχές μνημών (μνήμη Flash και SRAMs) έχουν δικαιώματα πρόσβασης μόνο για ανάγνωση/εγγραφή.

- Κατάσταση συστήματος: εκτέλεση της εφαρμογής, η εφαρμογή εκτελείτε (εκτελώντας πρώτα το ασφαλές τμήμα της εφαρμογής) μόλις επαληθεύσει η ασφαλής εκκίνηση ότι είναι εντάξει
 - Περιβάλλον εκτέλεσης: secure privileged, για την εκτέλεση του ασφαλούς privileged τμήματος της εφαρμογής (που αντιστοιχεί στο τμήμα PSA updatable RoT), και για την αποθήκευση δεδομένων που σχετίζονται με τις ασφαλείς υπηρεσίες SST και ITS.
 - Περιβάλλον εκτέλεσης: secure unprivileged, για την εκτέλεση του ασφαλούς unprivileged τμήματος της εφαρμογής (που αντιστοιχεί στο τμήμα updatable RoT).
 - Περιβάλλον εκτέλεσης: non-secure unprivileged (για την εκτέλεση του μη ασφαλούς τμήματος της εφαρμογής).

Το ασφαλές privileged τμήμα της εφαρμογής ξεκινά με την εκ νέου διαμόρφωση του συστήματος για να βάλει σε εφαρμογή τα προστατευμένα περιβάλλοντα εκτέλεσης που αναφέρονται παραπάνω και χρησιμοποιούνται κατά την εκτέλεση της εφαρμογής. Η επιφάνεια εκτέλεσης επεκτείνεται για όλο το ασφαλές μέρος. Μόλις ολοκληρωθεί η επαναδιαμόρφωση του συστήματος, το GTZC, η Secure MPU και η Secure SAU κλειδώνονται μέχρι την επόμενη επαναφορά ενεργοποιώντας τα bits κλειδώματος. Η μη ασφαλής εκτέλεση της εφαρμογής ξεκινά σε privileged λειτουργία και είναι σε θέση να ρυθμίσει εκ νέου τη μη ασφαλή MPU και να την κλειδώσει εάν χρειάζεται.

- WRP: Η προστασία εγγραφής (WRP) χρησιμοποιείται για την προστασία του αξιόπιστου κώδικα από εξωτερικές επιθέσεις ή ακόμα και από εσωτερικές τροποποιήσεις, όπως ανεπιθύμητες λειτουργίες εγγραφής/διαγραφής σε κρίσιμα σημεία κώδικα ή δεδομένα. Στο παράδειγμα TFM, το σύστημα έχει διαμορφωθεί ώστε να κάνει τον κώδικα εκκίνησης του TFM_SBSFU και τα εξατομικευμένα δεδομένα TFM_SBSFU_Boot ως αμετάβλητα δεδομένα.
- HDP: Όταν η προστασία HDP είναι ενεργοποιημένη, οποιαδήποτε πρόσβαση στην προστατευμένη περιοχή μνήμης flash (ανάκτηση, ανάγνωση, προγραμματισμός, διαγραφή) απορρίπτεται μέχρι την επόμενη επαναφορά του προϊόντος. Όλος ο κώδικας και τα μυστικά που βρίσκονται μέσα στην προστατευμένη περιοχή μνήμης flash είναι πλήρως κρυμμένα. Στο παράδειγμα TFM, το σύστημα έχει διαμορφωθεί ώστε να αποκρύπτει τον κώδικα εκκίνησης TFM_SBSFU_Boot, τα εξατομικευμένα δεδομένα TFM_SBSFU_Boot που βρίσκονται στη flash και τους TFM_SBSFU_Boot NV counters λίγο πριν από την εκκίνηση της επαληθευμένης εφαρμογής.
- Interrupts:
 - Κατά τη διάρκεια του TFM_SBSFU_Boot, όλα τα interrupt είναι απενεργοποιημένα εκτός από το NMI:
 - Secure vector table lock bit: Η διεύθυνση secure vector table μπορεί να κλειδωθεί μέχρι την επόμενη επαναφορά από το bit κλειδώματος ενεργοποίησης. Στο παράδειγμα TFM, η ασφαλής εφαρμογή κλειδώνει τον secure vector table κατά τη φάση προετοιμασίας. Η μη ασφαλής εφαρμογή μπορεί να κλειδώσει τον non secure vector table εάν χρειάζεται.

2.6 Bluetooth Low Energy

Το Bluetooth Low Energy (BLE) είναι μια τεχνολογία ασύρματης επικοινωνίας που λειτουργεί στη συχνότητα 2,4 GHz και χρησιμοποιεί χαμηλή κατανάλωση ενέργειας. Χρησιμοποιείται για επικοινωνία μικρής εμβέλειας μεταξύ συσκευών, όπως ένα smartphone με έναν tracker γυμναστικής ή ένας φορητό υπολογιστή με ένα πηκτρολόγιο. Το BLE χρησιμοποιεί μια τεχνολογία γνωστή ως "διαφήμιση" για τη μετάδοση μικρών πακέτων δεδομένων σε τακτά χρονικά διαστήματα, τα οποία άλλες

συσκευές BLE μπορούν να παραλάβουν και να διαβάσουν. Όταν δύο συσκευές BLE θέλουν να δημιουργήσουν μια σύνδεση, μπορούν να στείλουν και να λάβουν δεδομένα μέσω ενός πρωτοκόλλου "προσανατολισμένο στη σύνδεση". Το BLE μπορεί να υποστηρίξει πολλαπλές συνδέσεις ταυτόχρονα, γεγονός που το καθιστά χρήσιμο για συσκευές που αλληλεπιδρούν με πολλές άλλες συσκευές σε κοντινή απόσταση.[29]

2.6.1 Bluetooth vs BLE

Το Bluetooth και το BLE είναι και οι δύο τεχνολογίες ασύρματης επικοινωνίας που χρησιμοποιούνται για μεταφορά δεδομένων μικρής εμβέλειας μεταξύ συσκευών. Ωστόσο, διαφέρουν με διάφορους τρόπους:

- Κατανάλωση ενέργειας: Το BLE σχεδιάστηκε για να μειώνει την κατανάλωση ενέργειας και να αυξάνει τη διάρκεια ζωής της μπαταρίας, καθιστώντας το ιδανικό για συσκευές χαμηλής κατανάλωσης όπως φορητές συσκευές, αισθητήρες και συσκευές IoT. Το Bluetooth, από την άλλη πλευρά, καταναλώνει περισσότερη ενέργεια και είναι πιο κατάλληλο για συσκευές υψηλής ισχύος, όπως ηχεία και ακουστικά.
- Εμβέλεια: Το Bluetooth έχει μεγαλύτερη εμβέλεια από το BLE, το οποίο συνήθως περιορίζεται σε περίπου 10 μέτρα. Το Bluetooth μπορεί να φτάσει έως και 30 μέτρα ή περισσότερο, ανάλογα με την έκδοση.
- Ρυθμός δεδομένων: Το Bluetooth έχει υψηλότερο ρυθμό δεδομένων από το BLE, το οποίο σχεδιάστηκε για μικρές ποσότητες μετάδοσης δεδομένων. Το Bluetooth μπορεί να μεταδώσει δεδομένα έως και 2,1 Mbps, ενώ το BLE μπορεί να μεταδώσει έως και 1 Mbps.
- Συμβατότητα: Το Bluetooth χρησιμοποιείται ευρύτερα και υποστηρίζεται από μεγαλύτερο αριθμό συσκευών, ενώ το BLE χρησιμοποιείται κυρίως για συγκεκριμένες εφαρμογές όπως wearables, συσκευές υγειονομικής περίθαλψης και έξυπνες οικιακές συσκευές.

Συνοπτικά, το Bluetooth είναι πιο κατάλληλο για εφαρμογές υψηλού εύρους ζώνης όπως streaming μουσικής και βίντεο, ενώ το BLE είναι ιδανικό για εφαρμογές χαμηλής κατανάλωσης που απαιτούν μεγάλη διάρκεια μπαταρίας και μικρές ποσότητες μετάδοσης δεδομένων.[30]

2.6.2 ATT και GATT

Το ATT είναι συντομογραφία του Attribute Protocol. Χρησιμοποιείται για τον καθορισμό μιας ιεραρχικής δομής δεδομένων που αντιπροσωπεύει τις ιδιότητες μιας συσκευής Bluetooth. Αυτή η δομή δεδομένων χρησιμοποιείται για την ανταλλαγή πληροφοριών μεταξύ συσκευών Bluetooth. Το ATT είναι ένα πρωτόκολλο πελάτη-διακομιστή, όπου ο πελάτης στέλνει αιτήματα για ανάγνωση ή εγγραφή συγκεκριμένων χαρακτηριστικών και ο διακομιστής απαντά με τα ζητούμενα δεδομένα ή εκτελεί την ενέργεια που ζητήθηκε.

Το GATT σημαίνει Generic Attribute Profile. Είναι χτισμένο πάνω από το πρωτόκολλο ATT και ορίζει έναν τυπικό τρόπο ανταλλαγής δεδομένων μεταξύ συσκευών Bluetooth. Το GATT ορίζει ένα σύνολο υπηρεσιών, οι οποίες είναι συλλογές χαρακτηριστικών που εκτελούν μια συγκεκριμένη λειτουργία. Κάθε υπηρεσία έχει ένα μοναδικό UUID 16-bit (Universally Unique Identifier) που την προσδιορίζει. Το GATT ορίζει επίσης τους κανόνες για την πρόσβαση και την τροποποίηση των χαρακτηριστικών μιας υπηρεσίας.

Η κύρια διαφορά μεταξύ ATT και GATT είναι ότι το ATT ορίζει τη δομή δεδομένων που χρησιμοποιείται για την αναπαράσταση των χαρακτηριστικών μιας συσκευής Bluetooth, ενώ το GATT ορίζει τους κανόνες για την πρόσβαση και την τροποποίηση αυτών των χαρακτηριστικών. Με άλλα λόγια, το ATT είναι ένα πρωτόκολλο χαμηλότερου επιπέδου που καθορίζει τον τρόπο οργάνωσης και ανταλλαγής των δεδομένων, ενώ το GATT είναι ένα πρωτόκολλο υψηλότερου επιπέδου που ορίζει το περιεχόμενο και τη συμπεριφορά των δεδομένων.

Το ATT και το GATT είναι σημαντικά επειδή παρέχουν έναν τυποποιημένο τρόπο για τις συσκευές BLE να επικοινωνούν μεταξύ τους, διευκολύνοντας τη συνεργασία μεταξύ συσκευών διαφορετικών κατασκευαστών και έτσι οι συσκευές BLE μπορούν να χρησιμοποιηθούν ανεξάρτητα από τον κατασκευαστή ή τη μάρκα.[31]

2.6.3 Advertising data

Στο BLE, τα advertising data αναφέρονται στα μικρά πακέτα πληροφοριών που μεταδίδονται από μια συσκευή BLE σε άλλες συσκευές BLE που βρίσκονται κοντά της. Αυτά τα δεδομένα μεταδίδονται χρησιμοποιώντας μια διαδικασία που ονομάζεται advertising και χρησιμεύει ως ένας τρόπος για τις συσκευές BLE να επικοινωνούν τη διαθεσιμότητα και την παρουσία τους σε άλλες κοντινές συσκευές.

Τα advertising data μπορεί να περιέχουν διάφορους τύπους πληροφοριών, όπως το όνομα της συσκευής, το επίπεδο μπαταρίας της, τις υπηρεσίες που παρέχει και άλλα. Η μορφή και το περιεχόμενο των διαφημιστικών δεδομένων ορίζονται από την Bluetooth SIG (Special Interest Group), τον οργανισμό που είναι υπεύθυνος για την ανάπτυξη και τη συντήρηση του προτύπου Bluetooth.

Με αυτόν τον τρόπο, τα δεδομένα διαφήμισης επιτρέπουν στις συσκευές BLE να ανακαλύπτουν η μία την άλλη και να δημιουργούν συνδέσεις χωρίς την ανάγκη ρητής σύζευξης ή μη αυτόματης διαμόρφωσης.

Services και Characteristics

Οι υπηρεσίες (services) και τα χαρακτηριστικά (characteristics) BLE είναι τα δομικά στοιχεία επικοινωνίας μεταξύ συσκευών BLE. Παρέχουν έναν τρόπο για τις συσκευές BLE να ανταλλάσσουν δεδομένα με τυποποιημένο και καλά καθορισμένο τρόπο.

Μια "υπηρεσία" BLE είναι μια συλλογή σχετικών χαρακτηριστικών που παρέχουν μια συγκεκριμένη λειτουργία. Για παράδειγμα, μια Υπηρεσία καρδιακών παλμών μπορεί να περιλαμβάνει χαρακτηριστικά για τον τρέχοντα καρδιακό ρυθμό, τον ελάχιστο και μέγιστο καρδιακό ρυθμό και τη θέση του αισθητήρα καρδιακών παλμών. Οι υπηρεσίες ορίζονται από την Bluetooth SIG και προσδιορίζονται από ένα μοναδικό 128-bit UUID (Universally Unique Identifier).[32]

Ένα "χαρακτηριστικό" BLE είναι ένα μεμονωμένο κομμάτι δεδομένων που μπορεί να διαβαστεί ή να γραφτεί από μια συσκευή BLE. Τα χαρακτηριστικά ορίζονται μέσα σε μια υπηρεσία και αντιπροσωπεύουν μια συγκεκριμένη πληροφορία που μπορεί να ανταλλάσσεται μεταξύ συσκευών BLE. Κάθε χαρακτηριστικό έχει πολλές ιδιότητες, όπως τον τύπο του (π.χ. συμβολοσειρά, ακέραιος, boolean), το μέγεθός του και αν μπορεί να διαβαστεί, να γραφτεί ή να ειδοποιηθεί.

Οι υπηρεσίες και τα χαρακτηριστικά BLE παίζουν βασικό ρόλο στην επικοινωνία μεταξύ των συσκευών BLE. Παρέχουν έναν τρόπο για τις συσκευές BLE να ανταλλάσσουν δεδομένα με τυποποιημένο και καλά καθορισμένο τρόπο, διευκολύνοντας τις συσκευές διαφορετικών κατασκευαστών να συνεργάζονται. Για παράδειγμα, ένα έξυπνο ρολόι με δυνατότητα BLE που μετρά τον καρδιακό ρυθμό μπορεί να εκθέσει τα δεδομένα του καρδιακού παλμού του ως χαρακτηριστικό σε μια υπηρεσία καρδιακών παλμών, τα οποία μπορούν να διαβαστούν από άλλες συσκευές BLE, όπως ένα smartphone.[33]

Ιδιότητες Characteristic

Κάθε χαρακτηριστικό Bluetooth Low Energy (BLE) έχει τρεις ιδιότητες που καθορίζουν τον τρόπο πρόσβασης και χρήσης του από άλλες συσκευές BLE: "read", "write" και "notify". Αυτές οι ιδιότητες καθορίζουν τον τύπο επικοινωνίας που είναι δυνατή μεταξύ των συσκευών.

read Ένα χαρακτηριστικό "ανάγνωσης" μπορεί να διαβαστεί από μια άλλη συσκευή BLE, πράγμα που σημαίνει ότι η συσκευή μπορεί να ανακτήσει την τιμή του χαρακτηριστικού από τη συσκευή που το εκθέτει. Για παράδειγμα, ένα έξυπνο ρολόι με δυνατότητα BLE που μετρά τον καρδιακό ρυθμό μπορεί να έχει ένα χαρακτηριστικό για τον τρέχοντα καρδιακό ρυθμό, το οποίο μπορεί να διαβαστεί από ένα smartphone για να ανακτήσει την πιο πρόσφατη μέτρηση καρδιακού παλμού.

write Ένα χαρακτηριστικό "write" μπορεί να εγγραφεί από μια άλλη συσκευή BLE, πράγμα που σημαίνει ότι η συσκευή μπορεί να αλλάξει την τιμή του χαρακτηριστικού στη συσκευή που το εκθέτει. Για παράδειγμα, ένας λαμπτήρας με δυνατότητα BLE μπορεί να έχει ένα χαρακτηριστικό για το χρώμα του, στο οποίο μπορεί να γραφτεί ένα smartphone για να αλλάξει το χρώμα του φωτός.

notify Ένα χαρακτηριστικό "ειδοποίησης" μπορεί να στείλει ειδοποιήσεις σε μια άλλη συσκευή BLE όταν αλλάζει η τιμή της, πράγμα που σημαίνει ότι η συσκευή μπορεί να λαμβάνει ειδοποιήσεις κάθε φορά που αλλάζει η τιμή του χαρακτηριστικού. Για παράδειγμα, έναν αισθητήρα θερμοκρασίας BLE που μετρά περιοδικά τη θερμοκρασία περιβάλλοντος και εκπέμπει τη μέτρηση χρησιμοποιώντας ένα χαρακτηριστικό ειδοποίησης

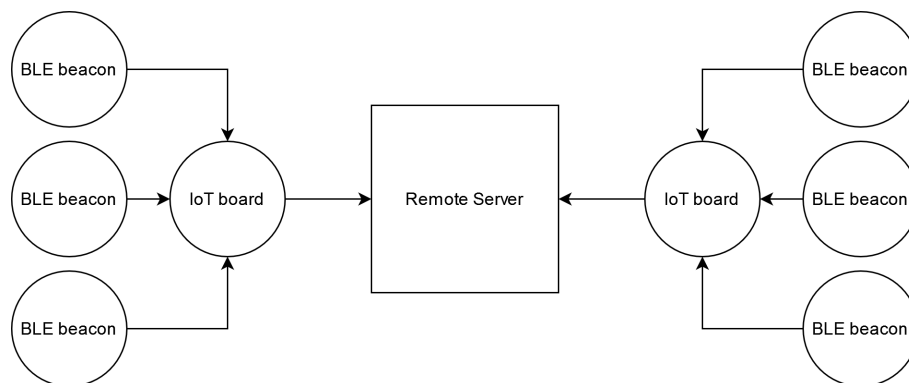
Αυτές οι ιδιότητες καθορίζουν τον τύπο επικοινωνίας που είναι δυνατή μεταξύ συσκευών BLE και διασφαλίζουν ότι τα δεδομένα ανταλλάσσονται με τυποποιημένο και καλά καθορισμένο τρόπο. Αυτό βοηθά να διασφαλιστεί ότι οι συσκευές BLE από διαφορετικούς κατασκευαστές μπορούν να συνεργάζονται και να ανταλλάσσουν δεδομένα με προβλέψιμο και δια λειτουργικό τρόπο.

ΚΕΦΑΛΑΙΟ 3

Υλοποίηση του Συστήματος

3.1 Αρχιτεκτονική συστήματος

Η εφαρμογή αποτελείται από πολλαπλούς ble beacons που στέλνουν δεδομένα σε πλακέτες IoT, οι οποίοι στη συνέχεια μεταδίδουν τα δεδομένα σε έναν κεντρικό διακομιστή για αποθήκευση. Κάθε ble beacon είναι μοναδικός και μπορεί να συλλέγει διαφορετικούς τύπους δεδομένων, όπως δεδομένα καιρού, δεδομένα τοποθεσίας ή δεδομένα αισθητήρων. Οι πλακέτες IoT λειτουργούν ως κόμβος για τα beacons, συλλέγοντας και επεξεργάζοντας τα δεδομένα πριν τα μεταδώσουν στον διακομιστή. Ο διακομιστής αποθηκεύει τα δεδομένα από όλους τις πλακέτες IoT και παρέχει πρόσβαση στα δεδομένα για ανάλυση σε τρίτο χρόνο.



Σχήμα 3.1: Αρχική ιδέα της εφαρμογής

3.1.1 Το δικό μας μοντέλο απειλών

Το πρώτο βήμα πριν τη σχεδίαση της αρχιτεκτονικής, είναι να δημιουργηθεί ένα μοντέλο απειλών 2.3.1 συγκεκριμένο για την εφαρμογή για να καθοριστούν καλύτερα οι απαιτήσεις ασφαλείας του συστήματος και έτσι να επιλεγεί το καλύτερο υλικό και λογισμικό.

Τρωτά σημεία Το ble beacon, η πλακέτα και ο διακομιστής θα μπορούσαν να έχουν κενά ασφαλείας τα οποία θα εκμεταλλευτεί ένας εισβολέας. Για παράδειγμα, το ble beacon έχει έναν αδύναμο αλγόριθμο κρυπτογράφησης ένας εισβολέας καταφέρνει να το σπάσει ή να συνδεθεί σε μη έμπιστη πλακέτα και να της στείλει τα δεδομένα, η πλακέτα να έχει έναν προεπιλεγμένο κωδικό πρόσβασης και να τον μαντέψει ένας εισβολέας ή ο διακομιστής θα μπορούσε να έχει μια παλιά έκδοση λογισμικού, με γνωστό τρωτό σημείο που έχει επιδιορθωθεί σε πιο πρόσφατη έκδοση και να το εκμεταλλευτεί ένας εισβολέας.

Τρόποι εκμετάλλευσης Ένας εισβολέας μπορεί να χρησιμοποιήσει διάφορους τρόπους για να εκμεταλλευτεί τα τρωτά σημεία στο beacon, την πλακέτα και τον διακομιστή. Για παράδειγμα, ένας εισβολέας μπορεί να εφαρμόσει μια επίθεση brute-force για να σπάσει την κρυπτογράφηση στο beacon, να προσποιηθεί ότι είναι μια έμπιστη πλακέτα έτσι ώστε το beacon να συνδεθεί με αυτή και να της στείλει τα δεδομένα ή να τροποποιήσει το λογισμικό της αληθινής έμπιστης πλακέτας για να διαβάσει τα δεδομένα στέλνοντας τα σε δικό του διακομιστή.

Επιτιθέμενοι Επιτιθέμενος είναι οποιοσδήποτε έχει τα κίνητρα και την τεχνική τεχνογνωσία να εκμεταλλευτεί τα τρωτά σημεία του συστήματος. Αυτό περιλαμβάνει χάκερς, εγκληματίες ή ακόμη και εργαζόμενους και εργολάβους με πρόσβαση στο σύστημα.

Απειλές Οι πιθανές απειλές για το σύστημα περιλαμβάνουν μη εξουσιοδοτημένη πρόσβαση, κλοπή, τροποποίηση των δεδομένων καιρού ακόμα και έλεγχο του ίδιου του ble beacon ή χειρότερα του συστήματος της ίδιας της πλακέτας που θα περιέχει περισσότερες ευαίσθητες πληροφορίες.

Ορισμένοι πιθανοί τρόποι επίθεσης που πρέπει να ληφθούν υπόψη κατά την ανάπτυξη ενός μοντέλου απειλής για αυτήν την εφαρμογή IoT:

- Φυσικές επιθέσεις: Ένας εισβολέας θα μπορούσε να αποκτήσει φυσική πρόσβαση στο ble beacon ή την πλακέτα μας και να τροποποιήσει ή να κλέψει τα δεδομένα ή και την ίδια τη συσκευή.
- Επιθέσεις Man-in-the-Middle: Ένας εισβολέας θα μπορούσε να υποκλέψει τα δεδομένα μεταξύ του ble beacon και της πλακέτας ή μεταξύ της πλακέτας και του διακομιστή και να τροποποιήσει ή να κλέψει τα δεδομένα.
- Επιθέσεις πλευρικού καναλιού: Ένας εισβολέας θα μπορούσε να χρησιμοποιήσει επιθέσεις πλευρικού καναλιού για να αποκτήσει πρόσβαση στο σύστημα. Για παράδειγμα, ένας εισβολέας χρησιμοποιώντας ανάλυση ισχύος ή ηλεκτρομαγνητική ανάλυση εξαγάγει το κλειδί κρυπτογράφησης από τον ble beacon ή την πλακέτα, επιτρέποντάς του να αποκρυπτογραφήσει τα δεδομένα ή να αποκτήσει πρόσβαση στο σύστημα.
- Επιθέσεις κακόβουλου λογισμικού: Ένας εισβολέας θα μπορούσε να μολύνει το board με κακόβουλο λογισμικό που θα του επέτρεπε να αποκτήσει πρόσβαση στο σύστημα ή να κλέψει δεδομένα.

Για την αντιμετώπιση αυτών των απειλών, ορισμένες στρατηγικές θα μπορούσαν να περιλαμβάνουν:

- Ισχυρή κρυπτογράφηση: Επιβεβαίωση ότι το ble beacon και η πλακέτα χρησιμοποιούν ισχυρούς αλγόριθμους κρυπτογράφησης για την προστασία των δεδομένων κατά τη μεταφορά.
- Ασφαλή κανάλια επικοινωνίας: Χρήση ασφαλών καναλιών επικοινωνίας, όπως SSL/TLS, για τη μετάδοση δεδομένων μεταξύ της πλακέτας και του διακομιστή.
- Ασφαλές υλικολογισμικό: Επιβεβαίωση ότι το υλικολογισμικό στην πλακέτα είναι ασφαλές και ότι δεν μπορεί να τροποποιηθεί εύκολα από τους εισβολείς.
- Ασφαλής υλοποίηση: Η υλοποίηση αλγορίθμων κρυπτογράφησης στο beacon και την πλακέτα είναι ασφαλής και ανθεκτική σε επιθέσεις πλευρικού καναλιού.

3.1.2 Ο συνδυασμός Λογισμικό-Υλικό που διαλέξαμε

Security framework Το PSA (Platform Security Architecture) 2.4.2 θεωρείται ένα από τα καλύτερα security frameworks για συσκευές IoT λόγω της ολοκληρωμένης προσέγγισής του στην ασφάλεια. Σε αντίθεση με άλλα security frameworks, το PSA σχεδιάστηκε ειδικά για συσκευές IoT, λαμβάνοντας υπόψη τις μοναδικές προκλήσεις ασφαλείας που αντιμετωπίζουν αυτές οι συσκευές. Παρέχει μια πολυεπίπεδη προσέγγιση ασφάλειας, με κάθε επίπεδο να αντιμετωπίζει μια διαφορετική πτυχή της ασφάλειας, όπως αληθινή ταυτότητα έμπιστης συσκευής, ενημερώσεις υλικολογισμικού και ασφαλή επικοινωνία. Το framework είναι επίσης ανοιχτού κώδικα, καθιστώντας εύκολο για τους προγραμματιστές να το ενσωματώσουν στα προϊόντα τους. Επιπλέον, έχει αναπτυχθεί από μερικές από τις κορυφαίες εταιρείες του κλάδου, όπως η Arm και η Microsoft, και έχει λάβει ευρεία υποστήριξη, καθιστώντας τη μια αξιόπιστη λύση. Συνολικά, το PSA παρέχει μια ισχυρή και επεκτάσιμη προσέγγιση στην ασφάλεια, καθιστώντας το την καλύτερη επιλογή για να ασφαλιστεί μια εφαρμογή.

PSA Level 2 συσκευές Μετά την ανάλυση και την σύγκριση των επιπέδων ασφαλείας του PSA αποφασίστηκε ότι το ιδανικό επίπεδο PSA που πρέπει να έχει η εφαρμογή είναι το δεύτερο. Ενώ υπάρχουν πολλοί μικροελεγκτές διαθέσιμοι στην αγορά που διαθέτουν πιστοποίηση PSA (Platform Security Architecture) Level 2, επιλέχθηκε το board STM32L562E-DK με τον μικροελεγκτή STM32L562QE της σειράς STM32 L5 για διάφορους λόγους.

Ο πρώτος λόγος είναι η διαθεσιμότητα ενός έτοιμου παραδείγματος TF-M για τα STM32 L5 MCUs που απλοποιεί την εφαρμογή ασφαλών εφαρμογών και εξοικονομεί χρόνο και προσπάθεια στους προγραμματιστές.

Ο δεύτερος λόγος είναι ότι το documentation είναι ολοκληρωμένο, καλά οργανωμένο και κατανοητό. Περιλαμβάνει datasheets, reference manuals, user manuals και application notes που παρέχουν λεπτομερείς πληροφορίες σχετικά με τις δυνατότητες, τις δυνατότητες και τη χρήση του MCU.

Ο τρίτος λόγος είναι το ενσωματωμένο Bluetooth Low Energy (BLE) που διατίθεται στο STM32L562E-DK που απλοποιεί την υλοποίηση BLE εφαρμογών, εξαλείφοντας την ανάγκη για εξωτερικά εξαρτήματα ή πρόσθετη προσπάθεια ανάπτυξης.

3.2 Προγραμματιστικά εργαλεία

3.2.1 STM32CubeIDE

Το STM32CubeIDE είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για τη δημιουργία και τον εντοπισμό σφαλμάτων εφαρμογών μικροελεγκτών STM32. Είναι ένα δωρεάν, απλό και ελαφρύ IDE που βασίζεται στο Eclipse και υποστηρίζει μια ποικιλία αλυσίδων εργαλείων ανάπτυξης, συμπεριλαμβανομένων των GCC και ARM. Διαθέτει πρόγραμμα επεξεργασίας κώδικα για τη δημιουργία και τον εντοπισμό σφαλμάτων κώδικα C/C++ καθώς και μια γραφική διεπαφή χρήστη (GUI) για τη ρύθμιση των περιφερειακών του μικροελεγκτή. Για να βοηθήσει τους προγραμματιστές στον εντοπισμό σφαλμάτων του κώδικά τους, το STM32CubeIDE προσφέρει επίσης μια ποικιλία εργαλείων εντοπισμού σφαλμάτων, όπως παρακολούθηση μεταβλητών σε πραγματικό χρόνο, breakpoints και εκτέλεση κώδικα βήμα προς βήμα.

3.2.2 STM32CubeProgrammer

Το STM32CubeProgrammer (γνωστό και ως STM32CubeProg) είναι ένα εργαλείο που χρησιμοποιείται για τον προγραμματισμό μικροελεγκτών STM32. Είναι ένα εργαλείο γραμμής εντολών (command-line tool) που επιτρέπει τον προγραμματισμό binary αρχείων σε μικροελεγκτές STM32, καθώς και την εκτέλεση διαφόρων άλλων λειτουργιών όπως ανάγνωση και εγγραφή στην εσωτερική μνήμη του μικροελεγκτή, επαλήθευση του λογισμικού και της ακεραιότητας της μνήμης του μικροελεγκτή. Το STM32CubeProgrammer μπορεί να χρησιμοποιηθεί για τον προγραμματισμό μικροελεγκτών STM32 μέσω μιας ποικιλίας διεπαφών επικοινωνίας, συμπεριλαμβανομένων των UART, USB και JTAG. Είναι ένα αυτόνομο εργαλείο που μπορεί να χρησιμοποιηθεί ανεξάρτητα από το ολοκληρωμένο περιβάλλον ανάπτυξης STM32CubeIDE (IDE). Το STM32CubeProgrammer είναι χρήσιμο για τον προγραμματισμό μικροελεγκτών STM32 σε περιβάλλοντα παραγωγής, καθώς και για προγραμματιστές που πρέπει να εκτελέσουν λειτουργίες χαμηλού επιπέδου στον μικροελεγκτή.

3.2.3 Βιβλιοθήκη STM32Cube_FW_L5

Η βιβλιοθήκη L5 είναι ένα σύνολο βιβλιοθηκών λογισμικού για την οικογένεια μικροελεγκτών STM32, που αναπτύχθηκε από την STMicroelectronics. Αυτές οι βιβλιοθήκες παρέχουν μια σειρά από λειτουργίες και εργαλεία για τον προγραμματισμό του

μικροελεγκτή STM32, συμπεριλαμβανομένων λειτουργιών για περιφερειακό έλεγχο, διαχείριση μνήμης και διεπαφές επικοινωνίας όπως I2C, SPI και USART. Η βιβλιοθήκη L5 έχει σχεδιαστεί για να διευκολύνει τους προγραμματιστές να δημιουργούν εφαρμογές για τον μικροελεγκτή STM32, παρέχοντας μια αφηρημένη διεπαφή υψηλού επιπέδου στο υλικό του μικροελεγκτή. Είναι συμβατό με το STM32CubeIDE και μπορεί να χρησιμοποιηθεί με διάφορες πλακέτες ανάπτυξης όπως το Nucleo ή το Discovery Kit.

3.2.4 Tera Term

Το Tera Term είναι ένα δωρεάν και ανοιχτού κώδικα πρόγραμμα εξομοιωτή τερματικού για τα Microsoft Windows. Έχει σχεδιαστεί για να μιμείται τη λειτουργικότητα ενός τερματικού υπολογιστή, επιτρέποντας στους χρήστες να επικοινωνούν με το λειτουργικό σύστημα ενός υπολογιστή ή άλλες συσκευές που διαθέτουν διεπαφή γραμμής εντολών. Χρησιμοποιείται ευρέως ως πρόγραμμα σειριακού τερματικού για την επικοινωνία με μικροελεγκτές όπως το STM32 χρησιμοποιώντας πρωτόκολλα όπως το UART μέσω σύνδεσης USB.

3.3 Σημαντικά Option Bytes

- TZEN: Αυτό το Option Byte χρησιμοποιείται για ενεργοποίηση ή απενεργοποίηση του TrustZone. Κάποια option byte θέλουν τη ρύθμιση αυτή ενεργοποιημένη για να εμφανιστούν στον STM32CubeProg.
- DBANK: Χωρίζει τη μνήμη σε ένα μόνο Bank των 128bit data read ή σε δυο Bank των 64bit το καθένα.
- Secure Hide Protection area (HDP): Είναι ένα πρόσθετο επίπεδο προστασίας, στην ασφαλή λειτουργία, που επιτρέπει, για παράδειγμα, την υλοποίηση μιας εφαρμογής ασφαλούς εκκίνησης.
- Readout protection (RDP): Είναι ένα παγκόσμιο flag προστασίας της μνήμης Flash απο εξωτερική πρόσβαση μέσω του JTAG το οποίο χωρίζεται σε 4 επίπεδα (0, 0.5, 1, 2) με το τελευταίο να είναι το περισσότερο ασφαλές (καθόλου πρόσβαση στη μνήμη) και το πρώτο το λιγότερο (πρόσβαση σε όλη τη μνήμη).

Σημείωση: Για το PSA Level 2 πρέπει το RDP Level να είναι τουλάχιστον 1.

- Write protection (WRP): Είναι 4 παραμετροποιήσιμες περιοχές (start - end) στη μνήμη η οποίες αποτρέπουν τυχαίες ή κακόβουλες λειτουργίες εγγράφης/διαγραφής.

3.4 Βασικές υλοποιήσεις

3.4.1 Υλοποίηση Trustzone σε STM32

Το "Hello, world!" είναι ένα απλό πρόγραμμα που χρησιμοποιείται συχνά για να εισάγει έννοιες προγραμματισμού σε αρχάριους. Είναι ένα πρόγραμμα που εμφανίζει το κείμενο "Hello, world!" στην οθόνη όταν εκτελείται. Σκοπός του προγράμματος είναι να επιδείξει τη βασική δομή και σύνταξη μιας γλώσσας προγραμματισμού και να διδάξει τη διαδικασία δημιουργίας και εκτέλεσης ενός προγράμματος στη γλώσσα αυτή. Το ισοδύναμο ενός "Hello, world!" για μικροελεγκτές αναφέρεται συχνά ως "Hello, Blinky" ή "Blinky" και είναι ένα απλό πρόγραμμα που αναβοσβήνει ένα LED. Αυτό το πρόγραμμα είναι παρόμοιο με το "Hello, world!" πρόγραμμα, αλλά αντί να εμφανίζει κείμενο σε μια οθόνη, χρησιμοποιεί το υλικό του μικροελεγκτή για να ελέγξει ένα LED, ανάβοντας και απενεργοποιώντας το επανειλημμένα.

Το Trustzone είναι μια λειτουργία ασφαλείας που επιτρέπει στον μικροελεγκτή να λειτουργεί σε δύο διαφορετικές λειτουργίες: μια μη ασφαλή λειτουργία και μια ασφαλή λειτουργία. Όταν ο μικροελεγκτής βρίσκεται σε ασφαλή λειτουργία, ορισμένοι πόροι και περιφερειακά είναι προσβάσιμα μόνο σε αξιόπιστο κώδικα, ενώ σε μη ασφαλή λειτουργία, είναι προσβάσιμοι σε όλους τους κωδικούς. Αυτό το παράδειγμα δείχνει τη χρήση trustzone σε μικροελεγκτή STM32L5 αλλάζοντας την κατάσταση δύο LED. Το ασφαλές LED10 εναλλάσσεται κάθε δευτερόλεπτο (*SECURE_IO_TOGGLE_DELAY*) και παραμένει αναμμένο σε περίπτωση σφάλματος στον ασφαλή κώδικα ενώ το μη ασφαλές LED9 εναλλάσσεται δύο φορές πιο γρήγορα (*NONSECURE_IO_TOGGLE_DELAY*) και παραμένει αναμμένο σε περίπτωση σφάλματος σε μη ασφαλή κώδικα.[34][35][36]

Setup

- Bank1 = Bank2 = 128 sectors

Option bytes		
DB256	<input checked="" type="checkbox"/>	Unchecked : 256Kb single Flash: contiguous address in bank1 Checked : 256Kb dual-bank Flash with contiguous addresses
DBANK	<input checked="" type="checkbox"/>	This bit can only be written when all protection (secure, PCROP, HDP) are disabled Unchecked : Single bank mode with 128 bits data read width Checked : Dual bank mode with 64 bits data
SRAM2_PE	<input checked="" type="checkbox"/>	SRAM2 parity check enable Unchecked : SRAM2 parity check enable Checked : SRAM2 parity check disable
SRAM2_RST	<input type="checkbox"/>	SRAM2 Erase when system reset Unchecked : SRAM2 erased when a system reset occurs Checked : SRAM2 is not erased when a system reset occurs
nSWBOOT0	<input checked="" type="checkbox"/>	Software BOOT0 Unchecked : BOOT0 taken from the option bit nBOOT0 Checked : BOOT0 taken from PH3/BOOT0 pin
nBOOT0	<input checked="" type="checkbox"/>	nBOOT0 option bit Unchecked : nBOOT0 = 0 Checked : nBOOT0 = 1
PA15_PUPEN	<input checked="" type="checkbox"/>	PA15 pull-up enable Unchecked : USB power delivery dead-battery enabled/ TDI pull-up deactivated Checked : USB power delivery dead-battery disabled/ TDI pull-up activated
TZEN	<input checked="" type="checkbox"/>	Global TrustZone security enable Unchecked : Global TrustZone security disabled Checked : Global TrustZone security enabled
HDP1EN	<input type="checkbox"/>	Hide protection first area enable Unchecked : No HDP area 1

Σχήμα 3.2: Option Bytes DBANK και TZEN

- SECWM1_PSTRT = 0x0 and SECWM1_PEND = 0x7F, που σημαίνει ότι και οι 128 σελίδες της Bank1 έχουν οριστεί ως ασφαλείς
- SECWM2_PSTRT = 0x1 and SECWM2_PEND = 0x0, που σημαίνει ότι καμία σελίδα της Bank2 δεν έχει οριστεί ως ασφαλής, επομένως η Bank2 δεν είναι ασφαλής

Secure Area 1			
Name	Value		
SECWM1_PSTRT	Value <input type="text" value="0x7f"/>	Address <input type="text" value="0x0803f800"/>	Start page of first secure area
SECWM1_PEND	Value <input type="text" value="0x0"/>	Address <input type="text" value="0x08000000"/>	End page of first secure area
Write Protection 1			
Secure Area 2			
Name	Value		
SECWM2_PSTRT	Value <input type="text" value="0x1"/>	Address <input type="text" value="0x08040800"/>	Start page of second secure area
SECWM2_PEND	Value <input type="text" value="0x0"/>	Address <input type="text" value="0x08040000"/>	End page of second secure area

Σχήμα 3.3: Option Bytes για τις ασφαλής και μη ασφαλής περιοχές

Ανάλυση κώδικα

Ένα από τα πρώτα πράγματα που συμβαίνουν είναι η εκτέλεση ενός αρχείου που ονομάζεται "startup_XXXX.s", το οποίο δημιουργεί το heap και το stack. Ως μέρος αυτής της διαδικασίας, καλείται η συνάρτηση "SystemInit" η οποία με τη σειρά της αρχικοποιεί τη SAU. Αυτό περιλαμβάνει τη διαμόρφωση ενός συνόλου καταχωρητών και άλλων ρυθμίσεων για να διασφαλιστεί ότι όλα τα περιφερειακά είναι ασφαλή, με την Bank1 να είναι ασφαλής και την Bank2 να μην είναι ασφαλής. Μετά την

ολοκλήρωση της συνάρτησης "SystemInit" και τη ρύθμιση της SAU και ορισμένων ρολογιών, το πρόγραμμα μεταβαίνει στη συνάρτηση "main".

Σημείωση: Οι περιοχές που ορίζει η SAU πρέπει να είναι οι ίδιες με αυτές που ορίζουν τα Option Byte αλλιώς το λογισμικό δε θα δουλέψει.

Η συνάρτηση "main", στην ασφαλή μεριά, με τη σειρά της αρχικοποιεί το HAL, το SystemClock και το GPIO. Όλα τα IO είναι από προεπιλογή ασφαλή οπότε πρέπει να ελευθερωθούν σε μη ασφαλή εκτός από το LED10 που θα ανάβει για να υποδεικνύει την ασφαλή μεριά.

```
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_ICACHE_Init();
/* USER CODE BEGIN 2 */
SecureInitIODone = 1;

/* All IOs are by default allocated to secure */
/* Release them all to non-secure except PG.12 (LED10) kept as secure */
__HAL_RCC_GPIOA_CLK_ENABLE();
__HAL_RCC_GPIOB_CLK_ENABLE();
__HAL_RCC_GPIOC_CLK_ENABLE();
__HAL_RCC_GPIOD_CLK_ENABLE();
__HAL_RCC_GPIOE_CLK_ENABLE();
__HAL_RCC_GPIOF_CLK_ENABLE();
__HAL_RCC_GPIOG_CLK_ENABLE();
__HAL_RCC_GPIOH_CLK_ENABLE();
HAL_GPIO_ConfigPinAttributes(GPIOA, GPIO_PIN_All, GPIO_PIN_NSEC);
HAL_GPIO_ConfigPinAttributes(GPIOB, GPIO_PIN_All, GPIO_PIN_NSEC);
HAL_GPIO_ConfigPinAttributes(GPIOC, GPIO_PIN_All, GPIO_PIN_NSEC);
HAL_GPIO_ConfigPinAttributes(GPIOD, GPIO_PIN_All, GPIO_PIN_NSEC);
HAL_GPIO_ConfigPinAttributes(GPIOE, GPIO_PIN_All, GPIO_PIN_NSEC);
HAL_GPIO_ConfigPinAttributes(GPIOF, GPIO_PIN_All, GPIO_PIN_NSEC);
HAL_GPIO_ConfigPinAttributes(GPIOG, (GPIO_PIN_All & ~(GPIO_PIN_12)), GPIO_PIN_NSEC);
HAL_GPIO_ConfigPinAttributes(GPIOH, GPIO_PIN_All, GPIO_PIN_NSEC);
```

Σχήμα 3.4: Όλα τα GPIO ρυθμισμένα ως μη ασφαλή εκτός από το LED10

Τέλος, καλείτε η "NonSecure_Init" η οποία είναι υπεύθυνη για την αρχικοποίηση της μη ασφαλούς λειτουργίας. Η συνάρτηση "main", στη μη ασφαλή μεριά, ρυθμίζει δύο callback συναρτήσεις, μια για το secure fault και μια για το global TrustZone controller error η οποίες θα καλεστούν από τον ασφαλή κώδικα σε περίπτωση σφάλματος. Στη συνέχεια αρχικοποιείτε το LED09 και η ροή πάει σε ατέρμων βρόχο. Πλέον οι callback συναρτήσεις "HAL_SYSTICK_Callback" σε ασφαλή και μη μεριά είναι υπεύθυνες να αναβοσβήνουν τα LED.

Σε περίπτωση σφάλματος σε οποιαδήποτε λειτουργία του κώδικα θα καλεστεί η "Error_Handler" συνάρτηση και το αντίστοιχο LED θα παραμείνει αναμμένο.

Secure world	Non Secure World
<pre> /** * @brief SYSTICK callback. * @retval None */ void HAL_SYSTICK_Callback(void) { if (SecureTimingDelay != 0U) { SecureTimingDelay--; } else { /* Toggle PG.12 (LED10) */ HAL_GPIO_TogglePin(GPIOG, GPIO_PIN_12); SecureTimingDelay = SECURE_IO_TOGGLE_DELAY; } } </pre>	<pre> /** * @brief SYSTICK callback. * @retval None */ void HAL_SYSTICK_Callback(void) { if (NonSecureTimingDelay != 0U) { NonSecureTimingDelay--; } else { /* Toggle PD.03 (LED9) */ HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_3); NonSecureTimingDelay = NONSECURE_IO_TOGGLE_DELAY; } } </pre>

Σχήμα 3.5: Οι callback συναρτήσεις "HAL_SYSTICK_Callback"

Secure World	Non Secure World
<pre> /** * @brief This function is executed in case of error occurrence. * @retval None */ void Error_Handler(void) { /* USER CODE BEGIN Error_Handler_Debug */ /* User can add his own implementation to report the HAL error return state */ /* Insure LED10 is configured */ if (SecureInitIOdone != 0) { MX_GPIO_Init(); } /* LED10 on */ HAL_GPIO_WritePin(GPIOG, GPIO_PIN_12, GPIO_PIN_RESET); /* Infinite loop */ while (1) { } /* USER CODE END Error_Handler_Debug */ } </pre>	<pre> /** * @brief This function is executed in case of error occurrence. * @retval None */ void Error_Handler(void) { /* USER CODE BEGIN Error_Handler_Debug */ /* User can add his own implementation to report the HAL error return state */ /* Insure LED9 is configured */ if (NonSecureInitIOdone != 0) { MX_GPIO_Init(); } /* LED9 on */ HAL_GPIO_WritePin(GPIOD, GPIO_PIN_3, GPIO_PIN_RESET); /* Infinite loop */ while (1) { } /* USER CODE END Error_Handler_Debug */ } </pre>

Σχήμα 3.6: Οι "Error_Handler" συναρτήσεις

3.4.2 Υλοποίηση TF-M σε STM32

Το παράδειγμα περιλαμβάνει τα αρχεία διαμόρφωσης και τα scripts που απαιτούνται για το compile και την εκτέλεση του TF-M σε μια συσκευή STM32L5, καθώς και μια προ ρυθμισμένη και προ μεταγλωττισμένη έκδοση του λογισμικού. Ένα δείγμα εφαρμογής που δείχνει πώς να δουλεύουν οι ασφαλείς υπηρεσίες TF-M, συμπεριλαμβανομένης της ασφαλούς εκκίνησης, της ασφαλούς αποθήκευσης και του ασφαλούς εντοπισμού σφαλμάτων, περιλαμβάνεται επίσης στο παράδειγμα.

Το παράδειγμα που παρέχει η ST για την εγκατάσταση του TF-M βρίσκεται σε αυτό το φάκελο **"/STM32Cube_FW_L5_V1.4.0/Projects/STM32L562E-DK/Applications/TFM/"** της βιβλιοθήκης L5 και όποιο μονοπάτι windows υπάρχει σε αυτή την υποενοότητα είναι συνέχεια αυτού του φακέλου.

Για την πλήρη εγκατάσταση του TF-M και την ασφάλεια που προσφέρει πρέπει να ακολουθηθούν τα παρακάτω βήματα:

1. Προετοιμασία της συσκευής STM32L5. Αυτό επιτυγχάνεται θέτωντας τα op-

tion bytes στην σωστή τιμή για να μπορέσει να γίνει η εγκατάσταση.

2. Compile λογισμικού. Η σειρά με την οποία πρέπει να δημιουργηθούν τα εκτελέσιμα αρχεία καθώς το ένα στάδιο επηρεάζει το επόμενο.
3. Προγραμματισμός λογισμικού στην εσωτερική μνήμη Flash και στην εξωτερική μνήμη Flash του μικροελεγκτή STM32L5.

1. Προετοιμασία της συσκευής STM32L5

Μέσα στον φάκελο **"/TFM/TFM_SBSFU_Boot/STM32CubeIDE"** είναι τοποθετημένο το script **"regression.sh"** το οποίο πρέπει να εκτελέσουμε.

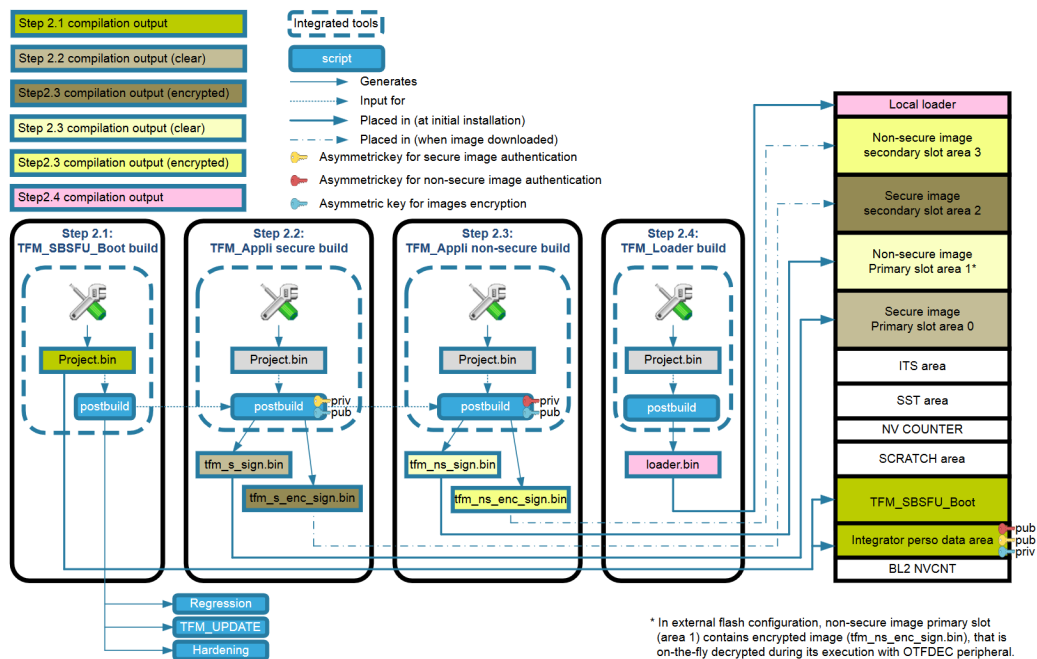
Το script χρησιμοποιεί το STM32 programmer command line interface (CLI) για την εκτέλεση ορισμένων ενεργειών. Το CLI χρησιμοποιείται έτσι ώστε το script να θέσει το RDP στο επίπεδο 0, να ενεργοποιήσει το TrustZone, και την κατάργηση των υπάρχουσών προστασιών για το bank 1 και το bank 2. Στη συνέχεια, η flash μνήμη διαγράφεται ολόκληρη και η προστασία HDP αφαιρείται. Το script εφαρμόζει τις προεπιλεγμένες επιλογές που είναι απαραίτητες για την εφαρμογή TF-M, τόσο στο bank 1 όσο και στο bank 2. Ένας άλλος τρόπος για να επιτευχθεί το ίδιο είναι χρησιμοποιώντας τη γραφική διεπαφή του STM32 programmer, η έξοδος και των δύο μεθόδων θα πρέπει να έχει ως αποτέλεσμα τα option byte να είναι όπως στον πίνακα 3.1.

Option Byte	Value
RDP	AA
SWAP_BANK	Unchecked
DBANK	Checked
SRAM2_RST	Unchecked
TZEN	Checked
HDP1EN	Unchecked
HDP1_PEND	Value 0x00, Address 0x08000000
HDP2EN	Unchecked
HDP2_PEND	Value 0x00, Address 0x08000000
SECBOOTADD0	Value 0x180052, Address 0x0c002900
SECWM1_PSTRT	Value 0x00, Address 0x08000000
SECWM1_PEND	Value 0x7f, Address 0x0803f800

Option Byte	Value
WRP1A_PSTRT	Value 0x7f, Address 0x0803f800
WRP1A_PEND	Value 0x00, Address 0x08000000
WRP1B_PSTRT	Value 0x7f, Address 0x0803f800
WRP1B_PEND	Value 0x00, Address 0x08000000
SECWM2_PSTRT	Value 0x00, Address 0x08040000
SECWM2_PEND	Value 0x7f, Address 0x0807f800
WRP2A_PSTRT	Value 0x7f, Address 0x0807f800
WRP2A_PEND	Value 0x00, Address 0x08040000
WRP2B_PSTRT	Value 0x7f, Address 0x0807f800
WRP2B_PEND	Value 0x00, Address 0x08040000

Πίνακας 3.1: Option Bytes Settings for TF-M

2. Compile λογισμικού



Σχήμα 3.7: Διαδικασία Compile TF-M

Η διαδικασία compile εκτελείται σε τέσσερα βήματα, όπως υποδεικνύεται στο σχήμα 3.7, λόγω των εξαρτήσεων μεταξύ αυτών των project, είναι σημαντικό να γίνουν build ακολουθώντας αυστηρά τη σειρά που περιγράφεται.

- Βήμα 1: Build TFM_SBSFU_Boot. Μερικές επιλογές διαμόρφωσης είναι διαθέσιμες, πριν από το Build, στο αρχείο **"boot_hal_cfg.h"** στο φάκελο **"/TFM/TFM_SBSFU_Boot/Inc/"**. Αυτό το βήμα δημιουργεί το δυαδικό αρχείο Secure Boot και Secure Firmware Update, συμπεριλαμβανομένων των παρεχόμενων δεδομένων χρήστη (keys, IDs κπλ.). Στο τέλος του build τρέχουν script τα οποία προετοιμάζουν άλλα script που χρειάζονται για το επόμενο βήμα.
- Βήμα 2: Build TFM_Appli secure. Αυτό το βήμα δημιουργεί το ασφαλές δυαδικό αρχείο TFM και χάρη σε μια εντολή postbuild που είναι ενσωματωμένη στο IDE project, παράγει το κρυπτογραφημένο και μη TFM secure signed image.
- Βήμα 3: Build TFM_Appli non secure. Αυτό το βήμα δημιουργεί το μη ασφαλές δυαδικό αρχείο TFM και χάρη σε μια εντολή postbuild που είναι ενσωματωμένη στο IDE project, παράγει το κρυπτογραφημένο και μη TFM non secure signed image.
- Βήμα 4: Build TFM_Loader. Αυτό το βήμα δημιουργεί την εικόνα TFM loader που θα γίνει flash στη συσκευή στα επόμενα βήματα.

3. Προγραμματισμός λογισμικού

Μέσα στον φάκελο **"/TFM/TFM_SBSFU_Boot/STM32CubeIDE"** είναι τοποθετημένο το script **"TFM_UPDATE.sh"** το οποίο πρέπει να εκτελέσουμε.

Το CLI χρησιμοποιείται πάλι για να προγραμματιστούν όλα τα δυαδικά αρχεία/εικόνες που δημιουργούνται στις μνήμες Flash. Η μορφή δεδομένων (καθαρή ή κρυπτογραφημένη) και η θέση της μνήμης Flash (εσωτερική μνήμη Flash ή εξωτερική μνήμη Flash) εξαρτώνται από τη διαμόρφωση του συστήματος που χρησιμοποιείται. Το script ενημερώνεται δυναμικά κατά το build του TFM_SBSFU_Boot, σύμφωνα με τη διάταξη της μνήμης Flash και σύμφωνα με τη διαμόρφωση της εφαρμογής, προκειμένου να διασφαλιστεί ότι τα δυαδικά αρχεία έχουν προγραμματιστεί στη σωστή θέση μνήμης Flash.

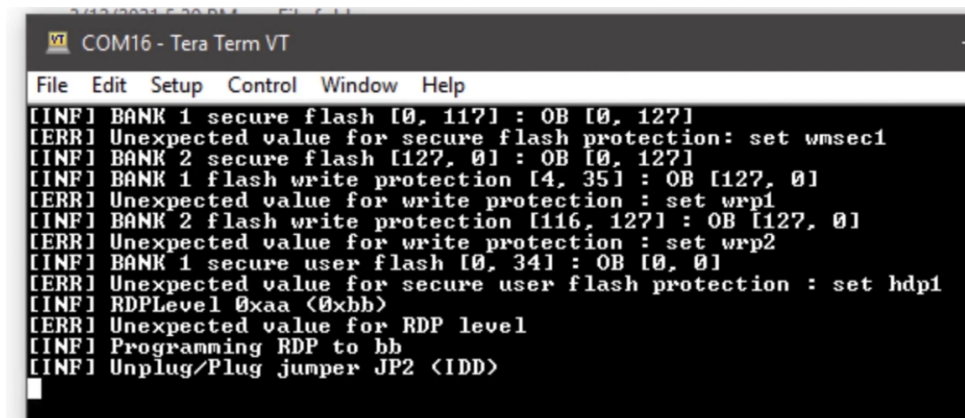
Πρώτη εκτέλεση

Απο το menu "Setup > Serial port..." οι ρυθμίσεις πρέπει να είναι ως εξής:

- Port: (COM Port του board απο τον device manager)

- Speed: 115200
- Data: 8 bit
- Parity: none
- Stop bits: 1 bit
- Flow control: none

Πατώντας το reset button Πάνω στο board το TFM_SBSFU_Boot ξεκινά. Στη συγκεκριμένη περίπτωση, στο αρχείο "**boot_hal_cfg.h**", το RDP επίπεδο είχε ρυθμιστεί στο 1 ενώ στα option bytes στο 0. Ανιχνεύεται μια εισβολή από τον μικροελεγκτή λόγω RDP επιπέδου 1, με αποτέλεσμα να παγώνει η εκτέλεση.



```

COM16 - Tera Term VT
File Edit Setup Control Window Help
[INF] BANK 1 secure flash [0, 117] : OB [0, 127]
[ERR] Unexpected value for secure flash protection: set wmsec1
[INF] BANK 2 secure flash [127, 0] : OB [0, 127]
[INF] BANK 1 flash write protection [4, 35] : OB [127, 0]
[ERR] Unexpected value for write protection : set wrp1
[INF] BANK 2 flash write protection [116, 127] : OB [127, 0]
[ERR] Unexpected value for write protection : set wrp2
[INF] BANK 1 secure user flash [0, 34] : OB [0, 0]
[ERR] Unexpected value for secure user flash protection : set hdp1
[INF] RDPLevel 0xaa (0xbb)
[ERR] Unexpected value for RDP level
[INF] Programming RDP to bb
[INF] Unplug/Plug jumper JP2 <IDD>

```

Σχήμα 3.8: Ανίχνευση εισβολής που εμφανίζεται στο Tera Term

Αφαιρώντας το JP2 jumper στην πλακέτα STM32L562E-DK και, στη συνέχεια, τοποθετώντας τον ξανά στη θέση του το board συνεχίζει μετά από την εισβολή.

Η εφαρμογή TFM_SBSFU_Boot ξεκινά με τις στατικές προστασίες που έχουν διαμορφωθεί σωστά. Στη συνέχεια, μεταβαίνει στο TFM_Appli εμφανίζοντας το κύριο μενού της εφαρμογής χρήστη στο Tera Term όπου δύο επιλογές είναι διαθέσιμες:

1. **“Test Protections”** Εκτελεί πρόσβαση στη μνήμη στην ασφαλή περιοχή από τη μη ασφαλή περιοχή. Φυσικά, όλες οι προσπάθειες πρέπει να αποτύχουν για να πετύχει το τεστ. (Πατώντας 1 και 1 ξανά στην επόμενη οθόνη)
2. **“Test TFM”** Εκτελεί τη δοκιμή ορισμένων ασφαλών υπηρεσιών TF-M. (Πατώντας 2 και 0 στην επόμενη οθόνη)

```
=====
<C> COPYRIGHT 2019 STMicroelectronics
=====
User App #A
=====

===== Main Menu =====
Test Protections ----- 1
Test TFM ----- 2
Selection :

```

Σχήμα 3.9: TF-M welcome screen display

Secure firmware update

Κρατώντας πατημένο το user button (μπλε) και απελευθερώνοντας το μετά το πάτημα του reset, ο χρήστης εισέρχεται στο local loader μενού. Ο local loader δεν είναι μέρος της μη ασφαλούς εφαρμογής TF-M, αλλά είναι μια αμετάβλητη αυτόνομη εφαρμογή, σε μη ασφαλή περιοχή.

```
=====
<C> COPYRIGHT 2020 STMicroelectronics
=====
LOCAL LOADER
=====

===== New Fw Download =====
Reset to trigger Installation ----- 1
Download Secure Image ----- 2
Download NonSecure Image ----- 3

```

Σχήμα 3.10: TFM local loader welcome screen

Χρησιμοποιώντας το μενού "File > Transfer > YMODEM > Send..." στο Tera Term διαλέγουμε το non secure ή το secure ή και τα δυο δυαδικά αρχεία που θέλουμε να στείλουμε στο board (tfm_ns_sign.bin, tfm_s_sign.bin). Αφού κατέβουν τα αρχεία, πατώντας το 1 το board κάνει επανεκκίνηση (ή πατώντας το reset button). Μετά την επανεκκίνηση, οι ληφθείσες εικόνες υλικολογισμικού εντοπίζονται, επαληθεύονται (συμπεριλαμβανομένου του ελέγχου κατά της επαναφοράς), αποκρυπτογραφούνται (εάν χρειάζεται), εγκαθίστανται και εκτελούνται από το TFM_SBSFU_Boot. [28][37][38]

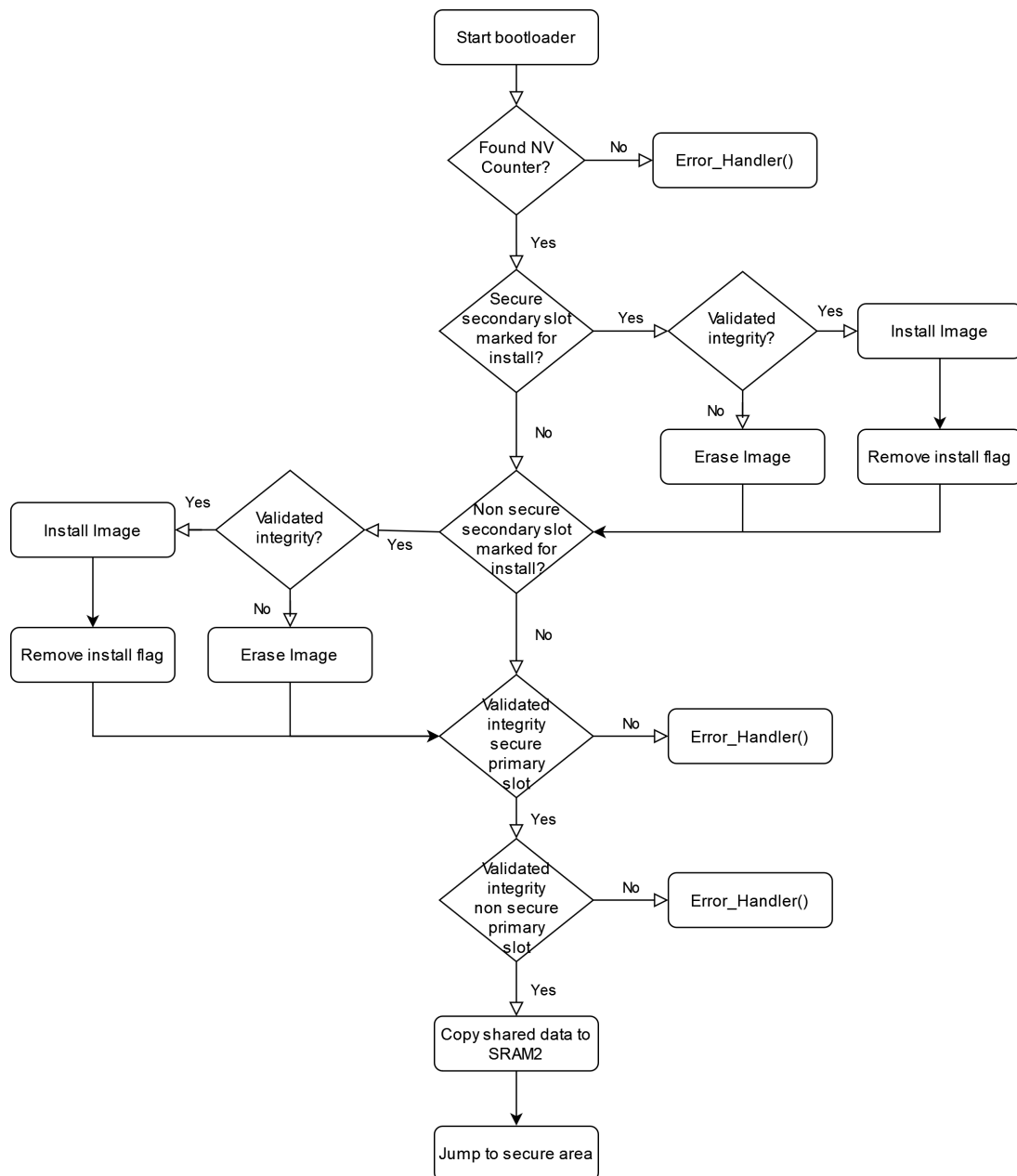
Σημείωση: Τα μονοπάτια στα script "**TFM_UPDATE.sh**" και "**regression.sh**", στο σύστημα μου, έπρεπε να τροποποιηθούν διότι ο STM32CubeProg ήταν εγκατεστημένος σε διαφορετική τοποθεσία από την προκαθορισμένη.

Αλλαγές στα κλειδιά

Στην εφαρμογή SBSFU του TF-M, από προεπιλογή, το σχήμα κρυπτογράφησης είναι η υπογραφή RSA-2048 και ο χρήστης πρέπει να αλλάξει τα κλειδιά διότι αυτά που έχει προεπιλεγμένα είναι ίδια για όλους όσους κατέβασαν το repo.2.6

- **Middleware/Third_Party/trustedfirmware/bl2/ext/mcuboot/**
 - RSA 2048 private key for secure image signature generation:
/root-rsa-2048.pem
 - RSA 2048 private key for non-secure image signature generation:
/root-rsa-2048_1.pem
 - RSA 2048 public key for AES-CTR key encryption:
/enc-rsa2048-pub.pem
- **"Middleware/Third_Party/trustedfirmware/bl2/ext/mcuboot/keys.c"**
 - RSA 2048 public Key for secure image signature verification:
rsa_pub_key
 - RSA 2048 public Key for non-secure image signature verification:
rsa_pub_key_1
 - RSA 2048 private key for AES-CTR key: decryptionenc_priv_key
- **"/TFM/TFM_SBSFU_Boot/Src/tfm_bl2_shared_data.c"**
 - HUK: huk_value
 - EAT private key: initial_attestation_curve_type,
initial_attestation_private_key, initial_attestation_private_key_size
 - EAT public key: initial_attestation_public_x_key,
initial_attestation_public_y_key
 - Instance ID: initial_attestation_raw_public_key_hash

3.5 Υλοποιήσεις ασφαλών υπηρεσιών στο TF-M



Σχήμα 3.11: Διάγραμμα ροής ασφαλούς εκκίνησης και ασφαλούς ενημέρωσης υλικολογισμικού

Στο παραπάνω διάγραμμα ροής περιγράφονται τα βήματα που ακολουθεί ο bootloader πριν παραδώσει τον έλεγχο στη secure περιοχή. Ελέγχει αν υπάρχουν NV Counters, αν χρειάζεται να γίνει αναβάθμιση και αν οι εγκατεστημένες εικόνες είναι αξιόπιστες. Πιο αναλυτικά περιγράφεται η διαδικασία στις επόμενες δύο υποενότητες:

3.5.1 Υλοποίηση Secure Boot

Το παρακάτω είναι log μετά από μια ασφαλής επαναφορά της συσκευής:

```
1 [INF] Starting bootloader
2 [INF] Checking BL2 NV area
3 [INF] Checking BL2 NV area header
4 [INF] Checking BL2 NV Counter consistency
5 [INF] Consistent BL2 NV Counter 3 = 0x1000000
6 [INF] Consistent BL2 NV Counter 4 = 0x1000000
7 [INF] Swap type: none
8 [INF] Swap type: none
9 [INF] verify counter 0 1000000 1000000
10 [INF] counter 0 : ok
11 [INF] verify sig key id 0
12 [INF] signature OK
13 [INF] verify counter 1 1000000 1000000
14 [INF] counter 1 : ok
15 [INF] verify sig key id 1
16 [INF] signature OK
17 [INF] Bootloader chainload address offset: 0x17000
18 [INF] Jumping to the first image slot
19 set to BL2 SHARED DATA2XX_HUK_CUSTOMIZATION_
20 [INF] Code c002900 c011c02
21 [INF] hash TFM_SBSFU_Boot 3721b2ee .. efa3896f
22 [INF] otfdec key ab, 62, 6e, f4, e6, b, f9, bc,
23 [INF] otfdec key a0, 94, 64, e1, ad, 42, f2, d9,
```

- Οι γραμμές 2-6 ανήκουν στη συνάρτηση *tfm_plat_init_nv_counter()* η οποία είναι υπεύθυνη να ελέγξει αν υπάρχουν οι NV Counters 3 και 4, και αν ναι, να διαβάσει τις τιμές τους.
- Οι γραμμές 7-8 ανήκουν στη συνάρτηση *boot_validated_swap_type()* η οποία είναι υπεύθυνη να ελέγξει αν υπάρχουν αλλαγές μεταξύ των ήδη εγκατεστημένων εικόνων και τον κατεβασμένων για αναβάθμιση. Στη συγκεκριμένη περίπτωση δεν υπάρχουν αλλαγές στις εικόνες.
- Οι γραμμές 9-12 ανήκουν στη συνάρτηση *bootutil_img_validate()* η οποία κάνει διάφορους ελέγχους στο image που της δόθηκε (Secure image index = 0) (**Μόνο τα headers των image διαβάζονται**)

- Οι γραμμές 9-10 αφορούν τον έλεγχο με *type == IMAGE_TLV_SEC_CNT* ο οποίος ελέγχει ότι η εικόνα έχει τουλάχιστον τον ίδιο security counter με τον NV Counter. (image_index 0, img_security_cnt 1000000, security_cnt 1000000)
- Οι γραμμές 11-12 αφορούν τον έλεγχο με *type == EXPECTED_SIG_TLV* ο οποίος ελέγχει ότι η υπογραφή της εικόνας είναι σωστή. (image_index 0, OK)
- Οι γραμμές 13-16 ανήκουν στη συνάρτηση *bootutil_img_validate()* η οποία κάνει διάφορους ελέγχους στο image που της δόθηκε (Non Secure image index = 1)
 - Οι γραμμές 13-14 αφορούν τον έλεγχο με *type == IMAGE_TLV_SEC_CNT* ο οποίος ελέγχει ότι η εικόνα έχει τουλάχιστον τον ίδιο security counter με τον NV Counter. (image_index 1, img_security_cnt 1000000, security_cnt 1000000)
 - Οι γραμμές 15-16 αφορούν τον έλεγχο με *type == EXPECTED_SIG_TLV* ο οποίος ελέγχει ότι η υπογραφή της εικόνας είναι σωστή. (image_index 1, OK)
- Η γραμμή 19 ανήκει στη συνάρτηση *TFM_BL2_CopySharedData()* η οποία μεταφέρει όλα τα απαραίτητα δεδομένα (BOOT SEED, Hardocded Lifecycle value, Attest key material, Implementation id) στην κοινή μνήμη της SRAM. Το "DATA2XX_HUK_CUSTOMIZATION_" που εκτυπώνεται είναι το default huk κλειδί που έρχεται με το TF-M ως string αναπαράσταση.
- Οι γραμμές 20-21 ανήκουν στη συνάρτηση *ComputeImplementationId()* η οποία υπολογίζει το implementation id και είναι εσωτερική της πάνω. Το id αυτό είναι το hash ολόκληρου του secure και non secure κόσμου. Στη γραμμή 20 φαίνεται απο που μέχρι που είναι ο κώδικας του οποίου το hash θα υπολογιστεί και στη γραμμή 21 φαίνεται η αρχή και το τέλος αυτού του hash.
- Οι γραμμές 22-23 ανήκουν στη συνάρτηση *otfdec_config()* η οποία ρυθμίζει ospi flash (εκεί που είναι το firmware) για on the fly decryption (OTFDEC). Το otfdec κλειδί εκτυπώνεται στις δύο γραμμές.

3.5.2 Υλοποίηση Secure Firmware Update

Το παρακάτω είναι log μετά από την επανεκκίνηση της συσκευής, πριν από την οποία μια καινούργια non secure εικόνα είχε τοποθετηθεί για αναβάθμιση:

```

1 | [INF] Starting bootloader
2 | [INF] Checking BL2 NV area
3 | [INF] Checking BL2 NV area header

```

```

4 [INF] Checking BL2 NV Counter consistency
5 [INF] Consistent BL2 NV Counter 3 = 0x1000000
6 [INF] Consistent BL2 NV Counter 4 = 0x1000000
7 [INF] Swap type: none
8 [INF] Swap type: test
9 [INF] verify counter 1 1000000 1000000
10 [INF] counter 1 : ok
11 [INF] verify sig key id 1
12 [INF] signature OK
13 [INF] Image upgrade secondary slot -> primary slot
14 [INF] Erasing the primary slot
15 [INF] Copying the secondary slot to the primary slot: 0x100000 bytes
16 [INF] verify counter 0 1000000 1000000
17 [INF] counter 0 : ok
18 [INF] verify sig key id 0
19 [INF] signature OK
20 [INF] verify counter 1 1000000 1000000
21 [INF] counter 1 : ok
22 [INF] verify sig key id 1
23 [INF] signature OK
24 [INF] ab, 62, 6e, f4, e6, b , f9 ,bc,
25 [INF] a0, 94, 64, e1, ad, 42 , f2 ,d9,
26 [INF] Bootloader chainload address offset: 0x17000
27 [INF] Jumping to the first image slot
28 set to BL2 SHARED DATA2XX_HUK_CUSTOMIZATION_
29 [INF] Code c002900 c011c02
30 [INF] hash TFM_SBSFU_Boot 3721b2ee .. efa3896f
31 [INF] otfddec key ab, 62, 6e, f4, e6, b, f9, bc,
32 [INF] otfddec key a0, 94, 64, e1, ad, 42, f2, d9,

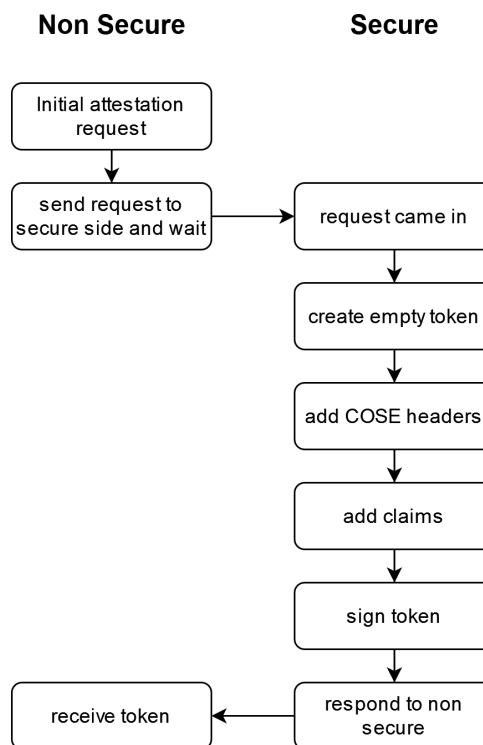
```

Παρατηρούμε ότι δεν υπάρχουν μεγάλες διαφορές σε σχέση με το log του απλού secure boot και έτσι δε θα αναλύσουμε αναλυτικά τις κοινές γραμμές με το secure boot 3.5.1.

- Οι γραμμές 2-7 παραμένουν ίδιες με τις αντίστοιχες το secure boot.
- Στη γραμμή 8 παρατηρούμε ότι εντοπίζεται αλλαγή στο non secure image (δεύτερος έλεγχος) και μαρκάρετε για τεστάρισμα.
- Στις γραμμές 9-12 πραγματοποιούνται οι έλεγχοι όπως στις αντίστοιχες του secure boot αλλά για το image που πρόκειται να εγκατασταθεί.

- Οι γραμμές 13-15 ανήκουν στη συνάρτηση `boot_copy_image()` η οποία είναι υπεύθυνη να αντιγράψει απο το secondary slot στο primary slot το image που της ζητήθηκε. Η συνάρτηση επίσης σβήνει το header απο το secondary slot έτσι ώστε στην επόμενη επανεκκίνηση να μην ξανά κάνει αναβάθμιση.
- Οι γραμμές 16-19 αφορούν τον έλεγχο του ήδη εγκατεστημένου secure image. (Γίνονται ξανά οι έλεγχοι όπως στις γραμμές 9-12)
- Οι γραμμές 20-23 αφορούν τον έλεγχο του πλέον ήδη εγκατεστημένου non secure image. (Γίνονται ξανά οι έλεγχοι όπως στις γραμμές 9-12)
- Οι γραμμές 24-25
- Οι γραμμές 25-32 παραμένουν ίδιες με τις 17-23 το secure boot.

3.5.3 Υλοποίηση Initial Attestation



Σχήμα 3.12: Διάγραμμα ροής Initial Attestation

Καλούμε τη συνάρτηση `psa_initial_attest_get_token()` για initial attestation με challenge:

a91b3c8d7e5f62442e987a136b77e18ff39a4d53c72c817bd0ae960eb59c175a88099b205dc5ba66
7f3a108c4e56e8996dc0f7297240b8223e4a3367906e252b

Παρακάτω φαίνεται ένα token πριν την κωδικοποίηση CBOR, με τους COSE headers και τα claims auth challenge, boot seed, instance id, implementation id, caller id, security lifecycle, all sw components και τέλος το hw version 2.5.3. Μπροστά από κάθε claim βρίσκεται ένας αριθμός που είναι το label του claim.

d243a10126a03a000124ff5840a91b3c8d7e5f62442e987a136b77e18ff39a4d53c72c817bd0ae96
0eb59c175a88099b205dc5ba667f3a108c4e56e8996dc0f7297240b8223e4a3367906e252b3a0001
24fb58203f7d2fccbe11dbd9885f9dedd3ce220994fa4cce8595015485c70c87c826c66f3a000125
00582101fa58755f658627ce5460f29b75296713248cae7ad9e2984b90280efcbcb502483a000124
fa58203721b2eeabceaba407bf05a9c60edc48a4becf76a3706abedef9a15bfa3896f3a000124f8
203a000124f91930003a000124fd82a501635350450465312e302e30025820a2934ee46d9f4fbcc3
e4f7309319feb05d4d00017f10778d62d3abcb4ba4c8920666534841323536055820fc5701dc6135
e1323847bdc40f04d2e5bee5833b23c29f93593d00018cfa9994a501644e5350450465312e302e30
0258201978a9a17eb86c406a80b6dd63dddecfcfdb86f4267a8faac5a55bf8206406930666534841
323536055820e18015993d6d2760b499274baef264b83af229e9a785f3d5bf00b9d32c1f03963a00
0124fc647264012000
00
00

Σημείωση: Η έννοια ή σημασία των αριθμών 5840, 5820 και 5821 ενδιάμεσα στα claims είναι ασαφής.

Μετά τη δημιουργία του token προστίθεται η υπογραφή και "κλείνει" η κωδικοποίηση CBOR. Το token επιστρέφεται στον χρήστη.

d28443a10126a0590193a83a000124ff5840a91b3c8d7e5f62442e987a136b77e18ff39a4d53c72c
817bd0ae960eb59c175a88099b205dc5ba667f3a108c4e56e8996dc0f7297240b8223e4a3367906e
252b3a000124fb582084d38dbbd2f97fe6d47488812765abd048376c1683c477b8a33271561fa56d
9a3a00012500582101fa58755f658627ce5460f29b75296713248cae7ad9e2984b90280efcbcb502
483a000124fa58203721b2eeabceaba407bf05a9c60edc48a4becf76a3706abedef9a15bfa3896f
3a000124f8203a000124f91930003a000124fd82a501635350450465312e302e3002582064380675
07e468e90e38eb30d32d5490218f0eb42c6c536c3dd155459f55ae630666534841323536055820fc
5701dc6135e1323847bdc40f04d2e5bee5833b23c29f93593d00018cfa9994a501644e5350450465
312e302e30025820cfbd84962ae3ad58769a67ff79a642111c6b331af4d83207fa00f812716cd81d
0666534841323536055820e18015993d6d2760b499274baef264b83af229e9a785f3d5bf00b9d32c
1f03963a000124fc647264012058400ecbc02c3ca38b278c3d3f5448f723aba6e36660b538f910b9
bfb674c2ef978ccb6577a12258e2f67bbe5fa8755453540e23c38fb8bdea492c722d72f518b366

ΚΕΦΑΛΑΙΟ 4

Μετρήσεις

4.1 Μετρήσεις χρόνου

Η εφαρμογή έχει τρεις διαφορετικές διαμορφώσεις διαθέσιμες. Η πρώτη λειτουργία έχει ενεργοποιημένα τόσο το TFM όσο και το Trustzone, η δεύτερη λειτουργία έχει ενεργοποιημένο μόνο το Trustzone και η τρίτη λειτουργία έχει και τις δύο λειτουργίες απενεργοποιημένες. Πραγματοποιήσαμε δοκιμές για τη μέτρηση του χρόνου εκκίνησης, του χρόνου λήψης δεδομένων και του χρόνου αποστολής δεδομένων για κάθε διαμόρφωση. Η σύγκριση αυτή είναι απαραίτητη για να κατανοήσουμε πόσο επηρεάζουν οι ρυθμίσεις αυτές την απόδοση του συστήματος.

Χρονομετρήσαμε επίσης τον χρόνο που χρειάζεται μια συσκευή με TF-M για initial attestation και για secure firmware update στην default διαμόρφωση του συστήματος.

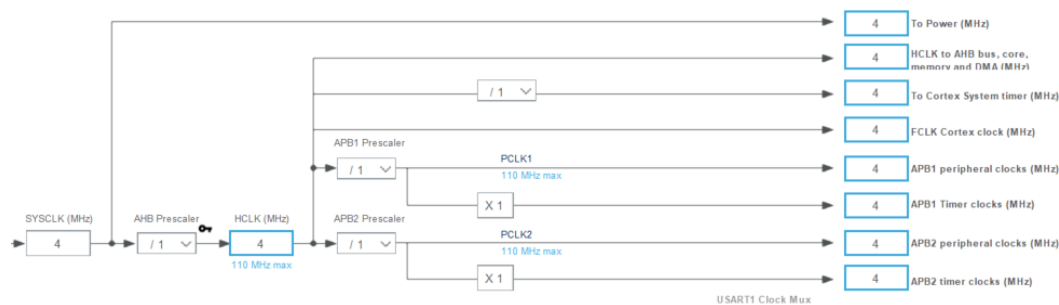
Σημείωση: Κατά τη μέτρηση των χρόνων λήφθηκε υπόψιν ότι το UART μπορεί να επηρεάσει τις μετρήσεις.

Σημείωση: Με System clock (SYSCLK) στα 110 MHz.

Timers: Το System clock (SYSCLK) μπορεί να διαιρεθεί με τη χρήση του AHB prescaler για να μας δώσει το clock signal για το AHB bus (HCLK). Αυτό με τη σειρά του μπορεί να διαιρεθεί με τον APB1 prescaler για να μας δώσει το APB1 clock (PCLK1) ή με τον APB2 prescaler για να μας δώσει το APB2 clock (PCLK2). Στα APB1 και APB2 bus βρίσκονται οι timers, με τον καθένα να έχει τον δικό του prescaler. Το STM32L562E-DK έχει 3 Low power timers, 1 RTC και 17 απλούς timers. Οι timers αυτοί είναι μετρητές που ανεβαίνουν με συχνότητα ανάλογη του

system clock και όλων των prescaler ενδιάμεσα. Οι TIM2, TIM3, TIM4, TIM5, TIM15, TIM16, TIM17 είναι General-purpose timers και θα χρησιμοποιούμε αυτούς.

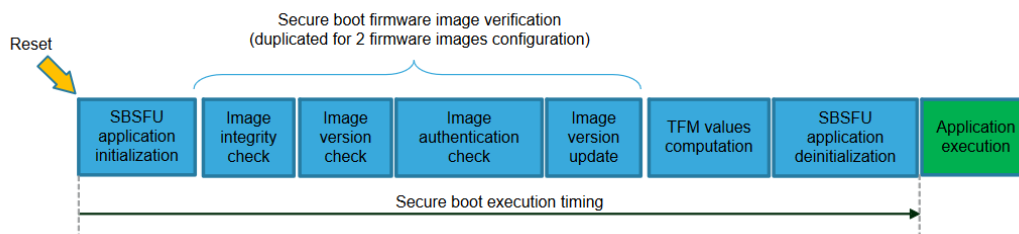
Χρησιμοποιώντας το σχήμα 4.1 ως παράδειγμα: Αν έχουμε SYSCLK στα 4MHz με όλους τους prescaler στο 1 και θέλουμε να μετρήσουμε ms (10^{-3} δευτερόλεπτα), πρέπει να θέσουμε τον prescaler του timer στα 4000 για να ανεβαίνει ο μετρητής του timer με συχνότητα 1KHz.



Σχήμα 4.1: STM32L562xx ρυθμίσεις ρολογιού

4.1.1 Χρόνος εκκίνησης

Θέλουμε να μετρήσουμε τον χρόνο που φαίνεται στο σχήμα 4.2 (το βελάκι). Όπως είδαμε και στην υλοποίηση του blinky με Trustzone 3.4.1 το πρώτο πράγμα που συμβαίνει σε κάθε επανεκκίνηση του συστήματος είναι να καλεστεί μια συνάρτηση με το όνομα SystemInit(). Μπορούμε να εκμεταλευτούμε αυτή τη συνάρτηση για να ξεκινήσουμε έναν timer 4.1 έτσι ώστε να μετρήσουμε τον χρόνο εκκίνησης (boot time). Τον timer τον σταματάμε σε όλες τις διαμορφώσεις αμέσως πριν την κλήση για αρχικοποίηση της εφαρμογής bluetooth.



Σχήμα 4.2: Secure Boot χρονοδιάγραμμα εκτέλεσης

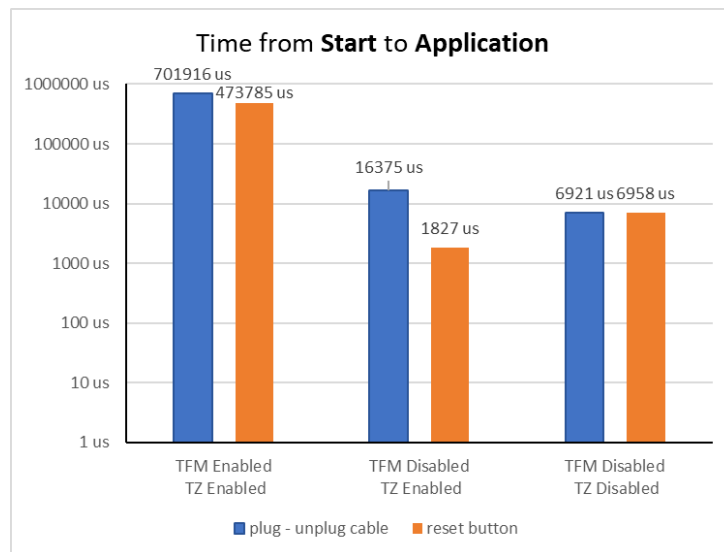
	TFM Enabled TZ Enabled	TFM Disabled TZ Enabled	TFM Disabled TZ Disabled
Εκκίνηση ⇒ Εφαρμογή	701.916 us	16.375 us	0 us
Εκκίνηση ⇒ NS main	554.207 us	125 us	X
Εκκίνηση ⇒ S main	450.967 us	0 us	X

Πίνακας 4.1: Boot time: plug - unplug cable

	TFM Enabled TZ Enabled	TFM Disabled TZ Enabled	TFM Disabled TZ Disabled
Εκκίνηση ⇒ Εφαρμογή	473.785 us	1.827 us	0 us
Εκκίνηση ⇒ NS main	412.125 us	125 us	X
Εκκίνηση ⇒ S main	308.904 us	0 us	X

Πίνακας 4.2: Boot time: reset button

Όταν αποσυνδεθεί και επανασυνδεθεί το καλώδιο, το board κάνει πλήρης επαναφορά που είναι διαφορετική από μια τυπική επαναφορά λογισμικού. Κατά τη διάρκεια μιας πλήρους επαναφοράς, όλοι οι καταχωρητές και τα περιφερειακά επαναφέρονται στην προεπιλεγμένη τους κατάσταση με αποτέλεσμα να διαρκεί περισσότερο από μια επαναφορά λογισμικού, η οποία επαναφέρει μόνο τον πυρήνα του επεξεργαστή.



Σχήμα 4.3: Σύγκριση χρόνων εκκίνησης

4.2 Μετρήσεις χρόνου συγκεκριμένες για το TF-M

4.2.1 Χρόνοι Secure Boot

Στο παρακάτω πίνακάκι "Προετοιμασία" θεωρείτε η διαδικασία ανάγνωσης της κεφαλίδας και ο καθορισμός του εάν χρειάζεται ενημέρωση ή όχι. "Επαλήθευση" θεωρείτε η διαδικασία επαλήθευσης που πραγματοποιεί ο bootloader πριν παραδώσει τον έλεγχο στην εφαρμογή.

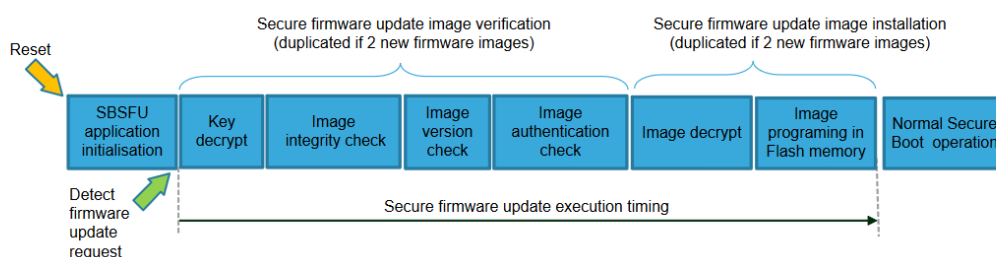
	Secure	Non Secure
Προετοιμασία	2.686 us	2.690 us
Επαλήθευση	43.027 us	44.108 us

Πίνακας 4.3: Χρόνοι "Προετοιμασίας" και "Επαλήθευσης" ασφαλούς εκκίνησης

4.2.2 Χρόνοι Secure Firmware Update

Όπως είδαμε και παραπάνω 3.5.1 πολλές διαδικασίες είναι παρόμοιες στο secure boot και στο secure firmware update. Μετρώντας τον χρόνο αυτών των διαδικασιών και στις δύο υπηρεσίες μας βοηθάει να τις συγκρίνουμε και να τις κατανοήσουμε καλύτερα.

Θα αναλύσουμε τώρα αυτές τις διαδικασίες που φαίνονται στο σχήμα 4.4:



Σχήμα 4.4: Secure Firmware Update χρονοδιάγραμμα εκτέλεσης

Το παρακάτω πίνακάκι αντιστοιχεί στο πρώτο μέρος (Image Verification) του σχήματος 4.4.

"Προετοιμασία" γίνεται όπως και στην ασφαλή εκκίνηση με μόνη διαφορά ότι τώρα υπάρχει εικόνα(ες) για εγκατάσταση. Η επαλήθευση αυτή είναι πιο χρονοβόρα από αυτή της ασφαλούς εκκίνησης διότι δε διαβάζονται και επαληθεύονται μόνο οι κεφαλίδες αλλά ολόκληρη η εικόνα.

Προετοιμασία / Επαλήθευση 1		
	Secure	Non Secure
Secure και Non-Secure	162.549 us	5.283.122 us
Μόνο Non-Secure	X	5.256.428 us
Μόνο Secure	162.545 us	X

Πίνακας 4.4: Χρόνοι "Προετοιμασίας" και "Επαλήθευσης 1" ασφαλούς ενημέρωσης

Τα δύο παρακάτω πίνακάκια αντιστοιχούν στο δεύτερο μέρος (Image Installation) του σχήματος 4.4.

Οι χρόνοι αυτοί είναι πόσο χρειάζεται για να διαγράψει την ήδη εγκατεστημένη εικόνα στο primary slot.

Εγκατάσταση (Διαγραφή παλιάς εικόνας)		
	Secure	Non Secure
Secure και Non-Secure	1.613.456 us	5.804.204 us
Μόνο Non-Secure	X	5.771.211 us
Μόνο Secure	1.613.455 us	X

Πίνακας 4.5: Χρόνοι εγκατάστασης (Διαγραφή παλιάς εικόνας) ασφαλούς ενημέρωσης

Οι χρόνοι αυτοί είναι πόσο χρειάζεται για να αντιγράψει την εικόνα για μαρκαισμένη για εγκατάσταση από το secondary στο primary slot.

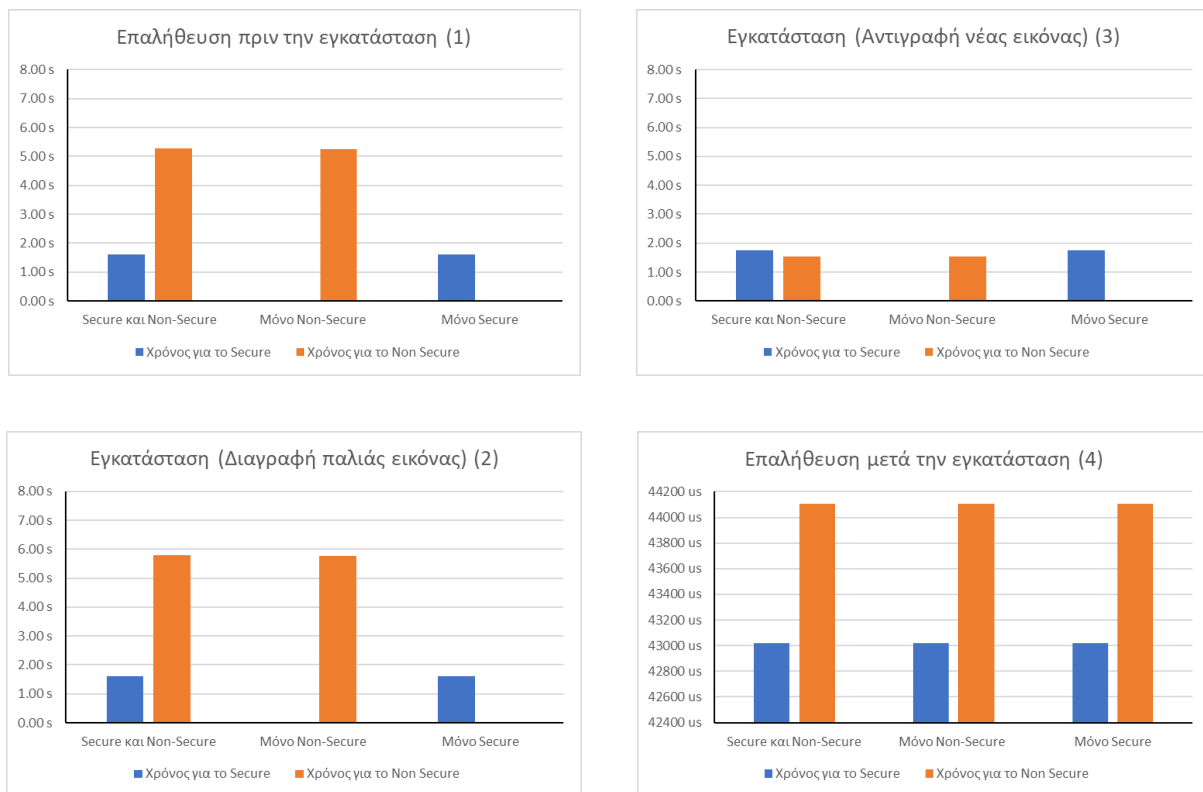
Εγκατάσταση (Αντιγραφή νέας εικόνας)		
	Secure	Non Secure
Secure και Non-Secure	1.752.736 us	1.533.526 us
Μόνο Non-Secure	X	1.534.858 us
Μόνο Secure	1.752.728 us	X

Πίνακας 4.6: Χρόνοι εγκατάστασης (Αντιγραφή νέας εικόνας) ασφαλούς ενημέρωσης

Οι χρόνοι αυτοί είναι η επαλήθευση που κάνει ο bootloader αφού έχει γίνει η εγκατάσταση και όπως βλέπουμε στο τέλος του σχήματος 4.4 η εκτέλεση βρίσκεται πλέον σε κανονική ασφαλή εκκίνηση.

Επαλήθευση 2 (Secure Boot Verify)		
	Secure	Non Secure
Secure και Non-Secure	43.021 us	44.107 us
Μόνο Non-Secure	43.020 us	44.105 us
Μόνο Secure	43.017 us	44.103 us

Πίνακας 4.7: Χρόνοι "Επαλήθευσης 2" ασφαλούς ενημέρωσης

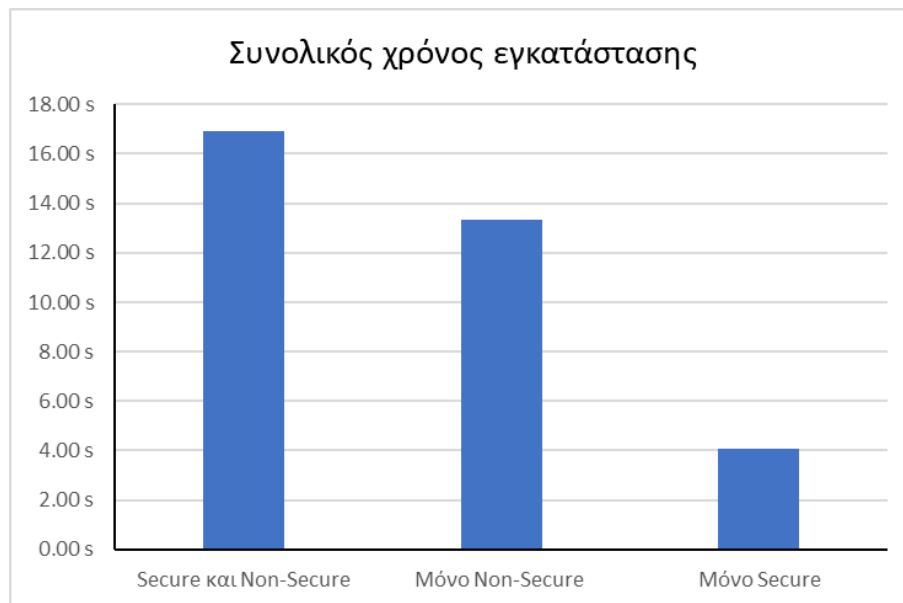


Σχήμα 4.5: Γραφήματα χρόνων Secure Firmware Update

Οι χρόνοι παρακάτω είναι αφού έχουμε κατεβάσει την εικόνα στη συσκευή και έχουμε πατήσει το reset (Εγκατάσταση image(s)):

Τύπος εικόνας	Συνολικός χρόνος εγκατάστασης
Secure και Non-Secure	16.945.877 us
Μόνο Non-Secure	13.340.848 us
Μόνο Secure	4.053.178 us

Πίνακας 4.8: Συνολικοί χρόνοι εγκατάστασης secure και non-secure εικόνων



Σχήμα 4.6: Γράφημα συνολικών χρόνων εγκατάστασης εικόνων

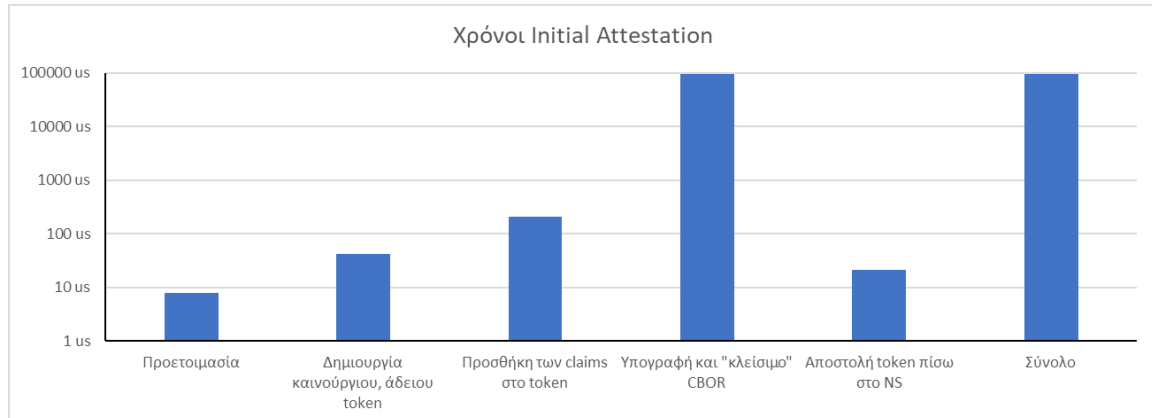
4.2.3 Χρόνοι Initial Attestation

Ο χρόνος αιτήματος initial attestation, κατασκευής token και απάντηση στον χρήστη (με HW acceleration) είναι **95.261 us**. Πιο αναλυτικά:

Περιγραφή	Χρόνος
Προετοιμασία, έλεγχοι (memcpy, έλεγχος μήκους challenge, κλπ.)	8 us
Δημιουργία καινούργιου, άδειου token	42 us
Προσθήκη των claims στο token 4.10	211 us
Υπογραφή και "κλείσιμο" CBOR	94.979 us

Περιγραφή	Χρόνος
Αποστολή token πίσω στο NS	21 us
Σύνολο	95.261 us

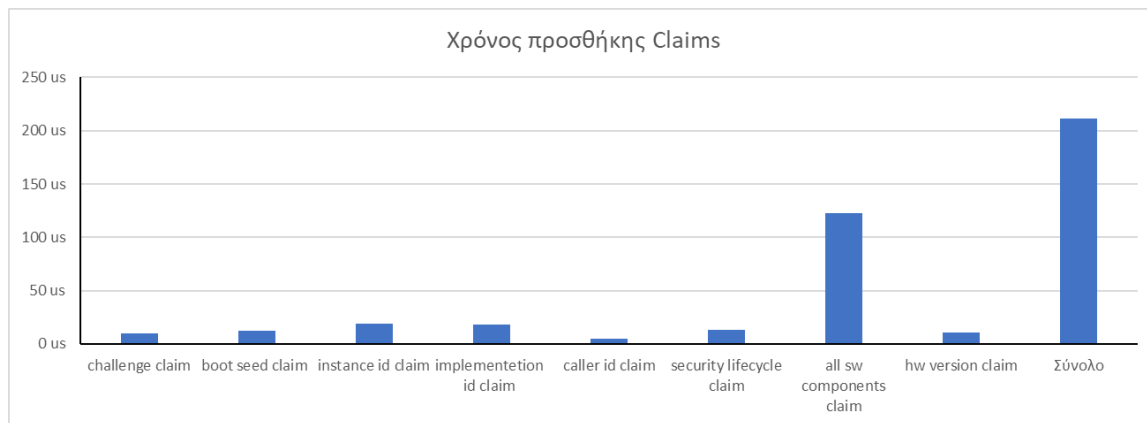
Πίνακας 4.9: Χρόνοι Initial Attestation



Σχήμα 4.7: Γράφημα χρόνων Initial Attestation

Όνομα claim	Χρόνος προσθήκης
challenge claim	10 us
boot seed claim	12 us
instance id claim	19 us
implementetion id claim	18 us
caller id claim	5 us
security lifecycle claim	13 us
all sw components claim	123 us
hw version claim	11 us

Πίνακας 4.10: Αναλυτικοί χρόνοι προσθήκης των claim



Σχήμα 4.8: Γράφημα αναλυτικών χρόνων προσθήκης των claim

4.3 Μετρήσεις κατανάλωσης ενέργειας

Κατά τη δοκιμή της κατανάλωσης ενέργειας του STM32, προσπαθήσαμε να βάλουμε την πλακέτα σε διαφορετικές λειτουργίες για να παρατηρήσουμε πώς η κατανάλωση ενέργειας ποικίλλει μεταξύ τους. Πρώτα θέτουμε την πλακέτα σε κανονική λειτουργία για να καθορίσουμε μια βασική μέτρηση κατανάλωσης ενέργειας. Στη συνέχεια προσπαθήσαμε να θέσουμε την πλακέτα σε κατάσταση αναστολής λειτουργίας, με ποικίλους τρόπους, όπου ο μικροελεγκτής περνά σε κατάσταση χαμηλής κατανάλωσης.

Μετρώντας την κατανάλωση ενέργειας της πλακέτας σε καθεμία από αυτές τις λειτουργίες, θα μπορέσουμε να προσδιορίσουμε ποια λειτουργία είναι η πιο αποδοτική από πλευράς ενέργειας για τη συγκεκριμένη περίπτωση χρήσης μας. Αυτές οι πληροφορίες θα μας επέτρεψαν να βελτιστοποιήσουμε τη χρήση ενέργειας της πλακέτας και να αυξήσουμε τη συνολική ενεργειακή της απόδοση.

Τροφοδοσία συσκευής Το STM32L562E-DK έχει σχεδιαστεί για να τροφοδοτείται κυρίως από πηγή ρεύματος 5V DC. Μπορεί να χρησιμοποιήσει μία από τις ακόλουθες εισόδους ισχύος, με κατάλληλη διαμόρφωση της πλακέτας:

1. **5V_STLK** Παρέχεται από έναν κεντρικό υπολογιστή συνδεδεμένο στο CN17 μέσω ενός Micro-B USB καλωδίου: 5V, 500mA
2. **5V_UCPD** Παρέχεται από έναν κεντρικό υπολογιστή συνδεδεμένο στο CN15 μέσω ενός USB Type-C καλωδίου: 5V, 1A maximum
3. **5V_VIN** Παρέχεται από έναν εξωτερικό τροφοδοτικό συνδεδεμένο στο CN18

pin 8: 7-12V, 800mA maximum

Για τις μετρήσεις συγκρίναμε μόνο τις δυο πρώτες 5V εισόδους χρησιμοποιώντας μια συσκευή ελέγχου θύρας USB 4.9 που μετράει την τάση και το ρεύμα εξόδου της.



Σχήμα 4.9: Συσκευή ελέγχου θύρας USB

Ισχύς Κανονικής λειτουργίας Μετρήσαμε την κατανάλωση ενώ η συσκευή βρίσκεται σε κανονική λειτουργία.

Ισχύς Εκκίνησης Όταν ο χρόνος εκκίνησης ενός MCU είναι απίστευτα γρήγορος, είναι δύσκολο να μετρηθεί με ακρίβεια η κατανάλωση ενέργειας κατά τη διαδικασία εκκίνησης. Για να ξεπεραστεί αυτό το πρόβλημα, μια λύση είναι η προσομοίωση της διαδικασίας εκκίνησης πολλές φορές μεταβαίνοντας ουσιαστικά στη διεύθυνση του bootloader κάθε φορά, αφού τελειώσει η εκκίνηση. Μετρώντας τον αριθμό των φορών που ξανά ξεκίνησε η εφαρμογή, η κατανάλωση ενέργειας μπορεί να μετρηθεί σε μεγαλύτερη κλίμακα, επιτρέποντας τον προσδιορισμό του μέσου όρου με μεγαλύτερη ακρίβεια. Θα χρησιμοποιήσουμε τη συνάρτηση `HAL_NVIC_SystemReset()` που κάνει επανεκκίνηση του συστήματος.

Ισχύς αναμονής

1. **while {1}** Η εντολή δημιουργεί έναν ατέρμονο βρόχο που δε σταματά ποτέ να εκτελείται.
2. **HAL_Delay(1000000)** Η συνάρτηση χρησιμοποιεί τον SysTick timer για να δημιουργήσει μια καθυστέρηση σε χιλιοστά του δευτερολέπτου. Το interrupt του SysTick timer αυξάνει μια παγκόσμια μεταβλητή που ονομάζεται `uwTick`, η συνάρτηση `HAL_Delay`

ελέγχει αυτή τη μεταβλητή και περιμένει μέχρι να φτάσει στην τιμή που έχει προκαθοριστεί.

3. **__WFI() (Wait For Interrupt)** Καλώντας τη συνάρτηση η συσκευή μπαίνει σε κατάσταση χαμηλής κατανάλωσης ενέργειας και περιμένει μέχρι να έρθει ένα interrupt (εξωτερικό ή εσωτερικό). Αν ποτέ δε στείλουμε αυτό το interrupt η συσκευή θεωρητικά παραμένει σε κατάσταση χαμηλής κατανάλωσης.
4. **Deep Sleep** Ρυθμίζοντας το SLEEPDEEP bit στο System Control Register (SCR) (SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk) και στη συνέχεια καλώντας τη συνάρτηση __WFI() ενεργοποιείται η λειτουργία βαθύ ύπνου.
5. **STOP 2** Σε αυτήν τη λειτουργία, ο πυρήνας και τα περισσότερα περιφερειακά είναι σταματημένα, και μόνο ορισμένα περιφερειακά μπορούν να αφυπνίσουν το σύστημα. Η διαφορά με τη λειτουργία βαθύ ύπνου είναι ότι επιτρέπει στον προγραμματιστή να επιλέξει ποια περιφερειακά να διατηρηθούν σε λειτουργία και ποια όχι ενώ η λειτουργία βαθύ ύπνου τα απενεργοποιεί όλα.

Στο TF-M παρατηρήθηκαν τα εξής δεδομένα: (Μέση τάση: 4,8V)

Μέτρηση	ST_Link Ένταση	Type-C Ένταση
Κανονική λειτουργία	0,23A	0,20A
Boot loop	0,20A	0,17A
while {1}	0,23A	0,19A
HAL_Delay(1000000)	0,22A	0,19A
__WFI()	0,22A	0,18A
Deep Sleep	0,18A	0,16A
STOP 2	0,19A	0,16A

Πίνακας 4.11: Αποτελέσματα μέτρησης έντασης TF-M

Απενεργοποιώντας το TF-M και έχοντας μόνο TrustZone: (Μέση τάση: 4,8V)

Μέτρηση	ST_Link Ένταση	Type-C Ένταση
Κανονική λειτουργία	0,20A	0,17A
Boot loop	0,20A	0,17A
while {1}	0,19A	0,17A
HAL_Delay(1000000)	0,18A	0,17A
__WFI()	0,19A	0,16A

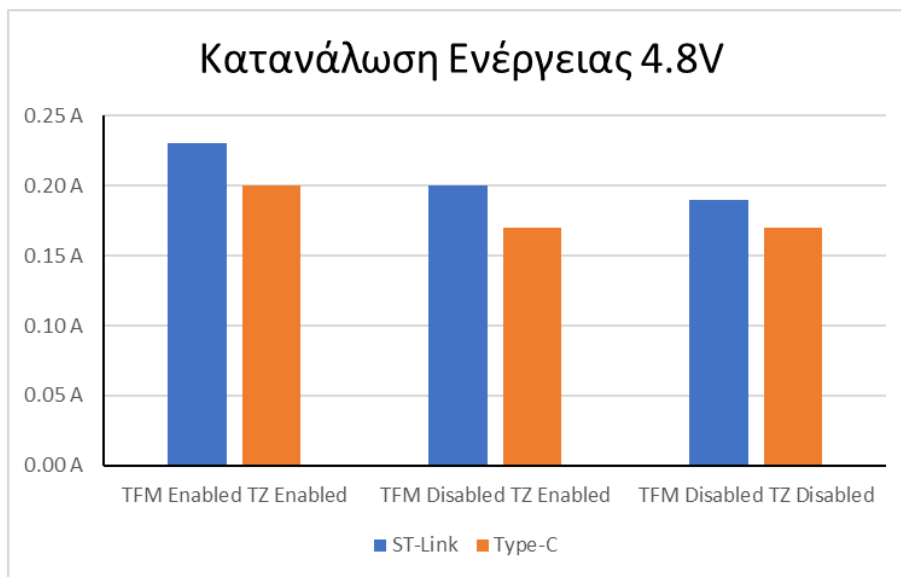
Μέτρηση	ST_Link Ένταση	Type-C Ένταση
Deep Sleep	0,19A	0,17A
STOP 2	0,18A	0,17A

Πίνακας 4.12: Αποτελέσματα μέτρησης έντασης TrustZone

Και απενεργοποιώντας και το TrustZone: (Μέση τάση: 4,8V)

Μέτρηση	ST_Link Ένταση	Type-C Ένταση
Κανονική λειτουργία	0,19A	0,17A
Boot loop	0,19A	0,16A
while {1}	0,19A	0,16A
HAL_Delay(1000000)	0,18A	0,17A
__WFI()	0,17A	0,16A
Deep Sleep	0,18A	0,17A
STOP 2	0,17A	0,16A

Πίνακας 4.13: Αποτελέσματα μέτρησης χωρίς ασφάλεια



Σχήμα 4.10: Γράφημα σύγκρισης έντασης σε κανονική λειτουργία

ΚΕΦΑΛΑΙΟ 5

Συμπεράσματα

5.1 Συμπεράσματα

Σε αυτή τη διατριβή, προτάθηκε ένα νέο σύστημα για ασφαλή μεταφορά δεδομένων που χρησιμοποιεί BLE beacons και το πλαίσιο ασφαλείας TrustZone με TF-M σε μια πλακέτα μικροελεγκτή STM32. Το σύστημα παρέχει ένα ασφαλές περιβάλλον για τη μεταφορά δεδομένων, διασφαλίζοντας την εμπιστευτικότητα και την ακεραιότητα των διαβιβαζομένων δεδομένων.

Αξιολογήθηκε η αποτελεσματικότητα του συστήματος όσον αφορά την ασφάλεια δεδομένων, την αξιοπιστία και την κατανάλωση ενέργειας. Τα αποτελέσματά δείχνουν ότι το σύστημα παρέχει υψηλά επίπεδα ασφάλειας και αξιοπιστίας δεδομένων, ενώ καταναλώνει σχετικά χαμηλά επίπεδα ενέργειας. Το TF-M παρέχει ισχυρή προστασία από ένα ευρύ φάσμα απειλών ασφαλείας, συμπεριλαμβανομένων των επιθέσεων λογισμικού και υλικού, και η χρήση του TrustZone ενισχύει τη συνολική ασφάλεια του συστήματος.

Οι μετρήσεις για τον χρόνο εκκίνησης και την κατανάλωση ενέργειας με και χωρίς ενεργοποιημένο TF-M και Trustzone παρέχουν πληροφορίες για την απόδοση του συστήματος σε διαφορετικές διαμορφώσεις. Ενώ ο χρόνος εκκίνησης αυξήθηκε ελαφρώς όταν ενεργοποιήθηκαν το TF-M και το TrustZone, ο συνολικός αντίκτυπος ήταν λογικός και αποδεκτός για τις περισσότερες εφαρμογές IoT. Είναι σημαντικό να σημειωθεί ότι τα οφέλη της βελτιωμένης ασφάλειας και προστασίας από πιθανές απειλές υπερτερούν κατά πολύ της αύξησης του χρόνου εκκίνησης. Ως εκ τούτου, συνίσταται η χρήση των TF-M και TrustZone σε εφαρμογές IoT όπου η ασφάλεια αποτελεί κορυφαία προτεραιότητα.

Συνολικά, το σύστημά προσφέρει μια πολλά υποσχόμενη λύση για μεταφορά δεδομένων σε εφαρμογές IoT, όπου η ασφάλεια και η αξιοπιστία είναι υψίστης σημασίας. Με την ανάπτυξη των εφαρμογών IoT, τα ασφαλή συστήματα μεταφοράς δεδομένων όπως αυτό

που προτείνεται σε αυτή τη διατριβή θα γίνονται όλο και πιο σημαντικά για τη διασφάλιση του απορρήτου και της ακεραιότητας των δεδομένων.

5.2 Μελλοντικές Επεκτάσεις

- Δικός μας Loader: Ο loader που έχει τώρα το πρότζεκτ είναι παράδειγμα για το πως να εγκαταστήσεις τις δυο εικόνες στο εκλεκτή. Μπορεί να φτιαχτεί ένας καλύτερος loader που θα επιτρέπει και απομακρυσμένη ασφαλή αναβάθμιση λογισμικού.
- Initial attestation: Αυτή τη στιγμή το laptop συνδέεται σε όποια συσκευή βρει και αρχίζει να δέχεται δεδομένα. Αυτό σημαίνει ότι οποιοσδήποτε χάκερ προσποιηθεί το board μας μπορεί να μας δώσει ψευδή δεδομένα. Η λύση είναι το λάπτοπ να ζητάει επιβεβαίωση ταυτότητας από το board πριν κρατήσει τα δεδομένα.
- Diffie - Hellman και κρυπτογράφηση δεδομένων: Ο Diffie - Hellman είναι ένας αλγόριθμος ανταλλαγής συμμετρικού κλειδιού. Θα μπορούσε να χρησιμοποιηθεί για να προστεθεί κρυπτογράφηση στα δεδομένα που στέλνονται, είτε stm32-laptop είτε beacon-stm32, έτσι ώστε και να υποκλαπούν από επιτεθειμένους να μην είναι χρήσιμα.
- Βελτίωση της επικοινωνίας Laptop - STM32: Η βιβλιοθήκη python που χρησιμοποιήθηκε είναι καινούργια και ακόμα όχι 100% λειτουργική. Επίσης, η διαδικασία μπορεί να απλουστευθεί περισσότερο αφαιρώντας τη χρήση BLE characteristics και services.
- Προσθήκη υποστήριξης για πολλαπλά ble beacons: Η εφαρμογή μας σκανάρει και δέχεται δεδομένα μόνο από ένα είδος beacon. Μια επέκταση θα ήταν η προσθήκη λογικής λήψης δεδομένων για διάφορες μάρκες και κατασκευαστές.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] M. Gudino, “Introduction to microcontrollers,” arrow.com. [Online]. Available: <https://www.arrow.com/en/research-and-events/articles/engineering-basics-what-is-a-microcontroller>
- [2] “Understanding flash: Blocks, pages and program / erases,” flashdba.com. [Online]. Available: <https://flashdba.com/2014/06/20/understanding-flash-blocks-pages-and-program-erases/>
- [3] L. Williams, “Difference between microprocessor and microcontroller,” guru99.com. [Online]. Available: <https://www.guru99.com/difference-between-microprocessor-and-microcontroller.html>
- [4] J. Koon, “Key factors to consider when choosing a microcontroller,” microcontrollertips.com. [Online]. Available: <https://www.microcontrollertips.com/key-factors-consider-choosing-microcontroller/>
- [5] A. S. Gillis, “What is the internet of things (iot)?” iotagenda. [Online]. Available: <https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT>
- [6] M. D. Forecast, “Global iot market | size, share, growth | 2022 to 2027,” 2021. [Online]. Available: <https://www.marketdataforecast.com/market-reports/internet-of-things-iot-market>
- [7] F. B. Insights, “Internet of things [iot] market size, share & trends, 2029,” 2021. [Online]. Available: <https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307>
- [8] Statista, “Iot market size worldwide 2017-2025 | statista,” 2019. [Online]. Available: <https://www.statista.com/statistics/976313/global-iot-market-size/>
- [9] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, “Massive internet of things for industrial applications: Addressing wireless iiot connectivity chal-

- lenges and ecosystem fragmentation,” *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 28–33, 2017.
- [10] G. Robinson, “Why cybersecurity has never been more important,” securitytoday.com. [Online]. Available: <https://securitytoday.com/Articles/2018/02/19/Why-Cybersecurity-Has-Never-Been-More-Important.aspx?Page=2>
- [11] A. Greenberg, “Hackers remotely kill a jeep on the highway—with me in it.” [Online]. Available: <https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/>
- [12] R. McMillan, “Siemens: Stuxnet worm hit industrial systems.” [Online]. Available: <https://www.computerworld.com/article/2515570/siemens--stuxnet-worm-hit-industrial-systems.html>
- [13] “What is embedded systems security?” windriver.com. [Online]. Available: <https://www.windriver.com/solutions/learning/embedded-systems-security>
- [14] McGraw-Hill, *Encyclopedia of Science and Technology*. McGraw-Hill Education, 2005.
- [15] S. Ravi, A. Raghunathan, and S. Chakradhar, “Tamper resistance mechanisms for secure embedded systems,” in *17th International Conference on VLSI Design. Proceedings*. IEEE, 2004, pp. 605–611.
- [16] N. Ouerdi, A. Azizi, and M. Azizi, “Classification of attacks on embedded systems,” *Journal of Computer Science*, 1834.
- [17] “Malware attacks: Definition and best practices,” rapid7.com. [Online]. Available: <https://www.rapid7.com/fundamentals/malware-attacks/>
- [18] E. on Tech, “What is a buffer overflow attack?” Youtube. [Online]. Available: <https://www.youtube.com/watch?v=YNkjX2Wqgh0>
- [19] “12 common attacks on embedded systems and how to prevent them,” apriorit.com. [Online]. Available: <https://www.apriorit.com/dev-blog/690-embedded-systems-attacks>
- [20] C. Shepherd, K. Markantonakis, N. van Heijningen, D. Aboukassim, C. Gaine, T. Heckmann, and D. Naccache, “Physical fault injection and side-channel attacks on mobile devices: A comprehensive analysis,” *IEEE Transactions on Information Forensics and Security*, 2015.
- [21] T. Group, “What is root of trust?” 2021. [Online]. Available: <https://cpl.thalesgroup.com/faq/hardware-security-modules/what-root-trust>

- [22] P. Certified, “What is a root of trust? | psa certified,” 2020. [Online]. Available: <https://www.psa-certified.org/blog/what-is-a-root-of-trust/>
- [23] R. Jadhav, “Memory protection unit.” [Online]. Available: <https://medium.com/system-level-solutions/memory-protection-unit-c3e61d651f14>
- [24] A. Mittal, “Introduction to cortex-m33/m25 sau/idaui.” [Online]. Available: <http://www.vlsiip.com/socsec/sau.html>
- [25] STMicroelectronics, “Secure boot and secure firmware update software expansion for stm32cube.” [Online]. Available: <https://www.st.com/en/embedded-software/x-cube-sbsfu.html>
- [26] —, “Overview of secure boot and secure firmware update solution on arm® trustzone® stm32 microcontrollers.” [Online]. Available: https://www.st.com/resource/en/application_note/an5447-overview-of-secure-boot-and-secure-firmware-update-solution-on-arm-trustzone-stm32-microcontrollers.pdf
- [27] T.-M. Org, “Secure boot.” [Online]. Available: https://tf-m-user-guide.trustedfirmware.org/design_docs/booting/tfm_secure_boot.html
- [28] STMicroelectronics, “Getting started with stm32cubel5 tfm application.” [Online]. Available: https://www.st.com/resource/en/user_manual/dm00678763-getting-started-with-stm32cubel5-tfm-application-stmicroelectronics.pdf
- [29] MUO, “What is ble (bluetooth low energy) and how does it work? - muo.” [Online]. Available: <https://www.makeuseof.com/what-is-ble-bluetooth-low-energy/>
- [30] R. W. World, “Bluetooth vs ble-difference between bluetooth and ble.” [Online]. Available: <https://www.rfwireless-world.com/Terminology/Bluetooth-vs-BLE.html>
- [31] R. D. Kevin Townsend Carles Cufí Akiba, “Getting started with bluetooth low energy.” [Online]. Available: <https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch04.html>
- [32] B. SIG, “16-bit uuid numbers document.” [Online]. Available: [https://btprodspecificationrefs.blob.core.windows.net/assigned-values/16-bit%20UUID%20Numbers%20Document.pdf](https://btprodspecificationrefs.blob.core.windows.net/assigned-values/16-bitUUIDNumbersDocument.pdf)
- [33] M. Afaneh, “Bluetooth gatt: How to design custom services and characteristics [midi device use case].” [Online]. Available: <https://novelbits.io/bluetooth-gatt-services-characteristics/>

- [34] STMicroelectronics, “Getting started with projects based on the stm32l5 series in stm32cubeide.” [Online]. Available: https://www.st.com/resource/en/application_note/an5394-getting-started-with-projects-based-on-the-stm32l5-series-in-stm32cubeide-stmicroelectronics.pdf
- [35] —, “Getting started with stm32l5 series microcontrollers and trustzone® development.” [Online]. Available: https://www.st.com/resource/en/application_note/an5421-getting-started-with-stm32l5-series-microcontrollers-and-trustzone-development-stmicroelectronics.pdf
- [36] —, “Stm32l5 mcu series using trustzone,” Youtube. [Online]. Available: <https://www.youtube.com/watch?v=cf3o9MT6bNI&list=PLnMKNibPkDnG6r7KMkC6H21Aq9Ke4MsUY&index=2>
- [37] —, “Stm32trust video series: Tfm,” Youtube. [Online]. Available: <https://www.youtube.com/watch?v=2svvqgSamJk&list=PLnMKNibPkDnEplEHaKh1SIH1Ulqnyr9kb&index=6>
- [38] “First run of the trusted firmware (tfm) application.” [Online]. Available: <https://blog.noser.com/first-run-of-the-trusted-firmware-tfm-application/>
- [39] STMicroelectronics, “How to disable trustzone in stm32l5xx devices during development phase.” [Online]. Available: https://wiki.st.com/stm32mcu/wiki/Security:How_to_disable_TrustZone_in_STM32L5xx_devices_during_development_phase
- [40] —, “Discovery kit with stm32l562qe mcu.” [Online]. Available: https://www.st.com/resource/en/user_manual/um2617-discovery-kit-with-stm32l562qe-mcu-stmicroelectronics.pdf
- [41] S. Community, “How to program option bytes with the hal api,” 2021. [Online]. Available: <https://community.st.com/s/article/how-to-program-option-bytes-with-the-HAL-API>

ΠΑΡΑΡΤΗΜΑ Α

Λήψη δεδομένων BLE απο MCU

A.1 nRF Connect App

Η εφαρμογή παρέχει μια σειρά λειτουργιών και εργαλείων που επιτρέπουν στους χρήστες να αναπτύξουν, να δοκιμάσουν και να εντοπίσουν σφάλματα Bluetooth Low Energy (BLE) και άλλες ασύρματες εφαρμογές χρησιμοποιώντας συσκευές nRF SoC. Με την εφαρμογή nRF Connect, οι χρήστες μπορούν να κάνουν σάρωση για κοντινές συσκευές Bluetooth, να συνδεθούν σε μια συγκεκριμένη συσκευή και να έχουν πρόσβαση στις υπηρεσίες και τα χαρακτηριστικά της. Η εφαρμογή περιλαμβάνει επίσης μια διαισθητική και φιλική προς το χρήστη γραφική διεπαφή για τη δημιουργία, τροποποίηση και δοκιμή υπηρεσιών και χαρακτηριστικών Bluetooth. Επιπλέον, η εφαρμογή επιτρέπει στους χρήστες να παρακολουθούν και να οπτικοποιούν δεδομένα σε πραγματικό χρόνο που ανταλλάσσονται μέσω Bluetooth, κάτι που είναι χρήσιμο για τον εντοπισμό σφαλμάτων και την αντιμετώπιση προβλημάτων ασύρματων εφαρμογών.

A.2 Python App

Η χρήση της Python για τη λήψη δεδομένων μέσω BLE στον υπολογιστή μπορεί να προσφέρει μια ευέλικτη και εύχρηστη λύση που μπορεί να προσαρμοστεί για να καλύψει τις περισσότερες ανάγκες. Με τη μεγάλη κοινότητα προγραμματιστών και το ευρύ φάσμα διαθέσιμων βιβλιοθηκών και εργαλείων, η Python είναι το κατάλληλο εργαλείο για να υλοποιήσει κάποιος γρήγορα και αποτελεσματικά την εφαρμογή που θέλει. Πολύ σημαντικό είναι η δυνατότητα αυτοματοποίησης ολόκληρης της διαδικασίας συλλογής και αποθήκευσης των δεδομένων.

A.2.1 Επιλογή βιβλιοθήκης

Η python έχει πολλές βιβλιοθήκες για BLE. Μερικές από τις πιο διάσημες είναι:

- bluepy - min Python version: 2.7, 3.4; supported OS: Linux
- pygatt - min Python version: 2.7, 3.3; supported OS: Linux, macOS, Windows
- gattlib - min Python version: 2.7, 3.3; supported OS: Linux
- bleak - min Python version: 3.6; supported OS: Windows, macOS, Linux, Android, iOS
- pybluez - min Python version: 2.6, 3.1; supported OS: Linux, macOS, Windows

Για την υλοποίηση της εργασίας επιλέχθηκε η bleak καθώς είναι η πιο σύγχρονη και υποστηρίζεται από τα windows.

A.2.2 Υλοποίηση

Αυτό το Python script A.2.2 έχει σχεδιαστεί για να διαβάζει δεδομένα θερμοκρασίας και υγρασίας από μια συσκευή Bluetooth χαμηλής ενέργειας (BLE) με συγκεκριμένη διεύθυνση και να καταγράφει αυτά τα δεδομένα σε ένα αρχείο CSV. Το script χρησιμοποιεί τις βιβλιοθήκες `asyncio` και `bleak` για σάρωση συσκευών BLE, σύνδεση στη συσκευή προορισμού και ανάγνωση των δεδομένων θερμοκρασίας και υγρασίας.

Το script ξεκινά ορίζοντας τις διευθύνσεις MAC των συσκευών BLE προς σάρωση, οι οποίες είναι οι διευθύνσεις του STM32, RHT beacon και BLU beacon. Στη συνέχεια ορίζεται η συνάρτηση `async def scan()`, η οποία χρησιμοποιεί τη συνάρτηση `BleakScanner.discover()` για σάρωση για συσκευές BLE. Εάν εντοπιστούν συσκευές, το script εκτυπώνει τις πληροφορίες της συσκευής. Οι συναρτήσεις `printResponse()` και `toString()` ορίζονται για τη μετατροπή των δεδομένων που λαμβάνονται από τη συσκευή BLE σε μορφή αναγνώσιμη από τον άνθρωπο.

Η συνάρτηση `async def update()` έχει οριστεί για σύνδεση σε μια συγκεκριμένη συσκευή BLE χρησιμοποιώντας τη διεύθυνσή της. Αφού συνδεθεί, η συνάρτηση διαβάζει τα δεδομένα θερμοκρασίας και υγρασίας από τη συσκευή χρησιμοποιώντας τη συνάρτηση `read_gatt_char()` και το προκαθορισμένο UUID των characteristics, μετατρέπει τα δεδομένα σε μορφή αναγνώσιμη από τον άνθρωπο και εγγράφει τα δεδομένα σε αρχείο CSV. Στη συνέχεια, η λειτουργία αποσυνδέεται από τη συσκευή.

Στη συνέχεια, το σενάριο ρυθμίζει το μονοπάτι αρχείου και τη κεφαλίδα για το αρχείο CSV. Εάν το αρχείο δεν υπάρχει, η σειρά κεφαλίδας εγγράφεται στο αρχείο. Ο βρόχος `while` στο τέλος του σεναρίου καλεί επανειλημμένα τη συνάρτηση `update()` για να διαβάσει και να γράψει τα δεδομένα θερμοκρασίας και υγρασίας στο αρχείο CSV. Ο βρόχος συνεχίζεται επ' αόριστον, επομένως το σενάριο θα εκτελείται έως ότου διακοπεί χειροκίνητα.

Συνολικά, το script έχει σχεδιαστεί για να διαβάζει συνεχώς δεδομένα θερμοκρασίας και υγρασίας από μια συσκευή BLE και να εγγράφει αυτά τα δεδομένα σε ένα αρχείο CSV για μεταγενέστερη ανάλυση.

```
1 from bleak import *
2 import time, os, csv, asyncio
3
4 STM_address = "F3:D9:EF:67:EB:81"
5 RHT_address = "C6:41:55:2C:DB:17"
6 BLU_address = "CC:78:AB:5E:78:A4"
7
8 def printResponse(uuid, response):
9     print(uuid, toString(response))
10
11 def toString(input):
12     return ''.join(format(x, '02x') for x in input)
13
14 async def update(address):
15     client = BleakClient(address)
16     while 1:
17         try:
18             await client.connect()
19             break
20         except Exception as e:
21             print(e)
22             time.sleep(5)
23
24     print("Connected to ", address)
25
26     # Capture the correct data using the char UUID
27     temperature = await client.read_gatt_char("00001f2a-0000-1000-8000-00805
        f9b34fb")
28     printResponse('temp ', temperature)
29     humidity = await client.read_gatt_char("00006f2a-0000-1000-8000-00805
        f9b34fb")
30     printResponse('hum ', humidity)
31
32     # Write the current time and data to the file
33     with open(file_path, 'a', newline='') as file:
34         writer = csv.writer(file)
35         current_time = time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
```

```

36         writer.writerow([current_time, toString(temperature), toString(humidity
           )])
37     await client.disconnect()
38
39 def createCSV():
40     # Get the directory of the script
41     script_directory = os.path.dirname(os.path.realpath(__file__))
42     # Define the path of the CSV file
43     file_path = os.path.join(script_directory, 'thesis_data.csv')
44     # Define the header row of the CSV file
45     header = ['Time', 'Temperature', 'Humidity']
46     # Write the header row to the file if the file does not already exist
47     if not os.path.exists(file_path):
48         with open(file_path, 'w', newline='') as file:
49             writer = csv.writer(file)
50             writer.writerow(header)
51
52 if __name__ == "__main__":
53     # Create the file to store all incoming data
54     createCSV()
55     # Start scanning for data
56     while(1):
57         asyncio.run(update(STM_address))

```


ΠΑΡΑΡΤΗΜΑ Β

X-CUBE-BLE1

Το πακέτο λογισμικού επέκτασης X-CUBE-BLE1 για STM32Cube εκτελείται στο STM32 και περιλαμβάνει προγράμματα οδήγησης για συσκευές χαμηλής κατανάλωσης Bluetooth BlueNRG-MS/BlueNRG-M0.

Η επέκταση βασίζεται στην τεχνολογία λογισμικού STM32Cube για να διευκολύνει τη φορητότητα σε διαφορετικούς μικροελεγκτές STM32.

B.1 Παράδειγμα SampleApp

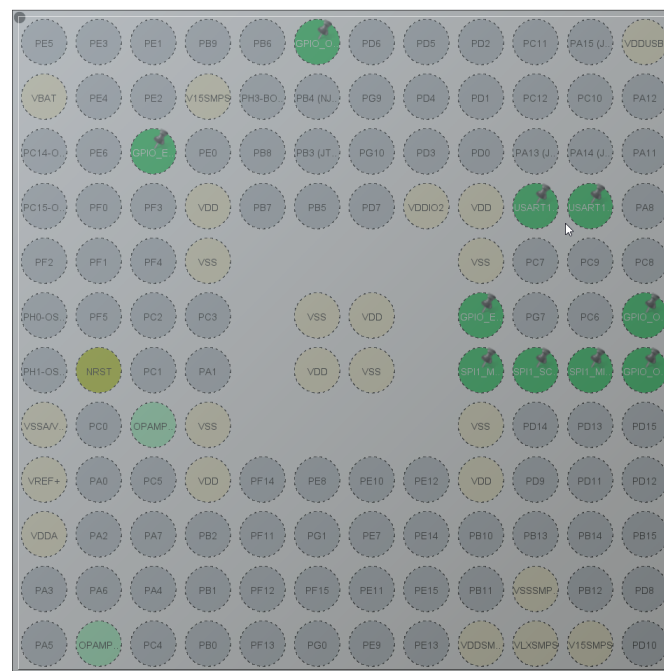
Ένα από τα παραδείγματα εφαρμογών που παρέχονται με το πακέτο X-CUBE-BLE1 είναι η εφαρμογή "SampleApp". Ο σκοπός αυτής της εφαρμογής είναι να δείξει πώς να χρησιμοποιείτε τη λειτουργικότητα BLE που παρέχεται από το πακέτο για τη δημιουργία μιας απλής ασύρματης σύνδεσης επικοινωνίας μεταξύ δύο συσκευών. Το SampleApp μπορεί να χρησιμοποιηθεί για τη δημιουργία σύνδεσης BLE μεταξύ μιας κεντρικής συσκευής (όπως ένα smartphone ή tablet) και μιας περιφερειακής συσκευής (όπως μια πλακέτα ανάπτυξης STM32) και την ανταλλαγή δεδομένων μεταξύ τους.

Η εφαρμογή "SampleApp" παρέχει στον χρήστη την δυνατότητα να ορίσει το ρόλο που θα έχει η συσκευή. Σε ρόλο "Server" η συσκευή δημιουργεί ένα Bluetooth Service και Characteristic 2.6 και περιμένει μια άλλη συσκευή να συνδεθεί για να της τα στείλει. Σε ρόλο "Client" η συσκευή ψάχνει μια άλλη συσκευή για να συνδεθεί και να διαβάσει τα δεδομένα που αυτή στέλνει.

Στις εικόνες παρακάτω φαίνονται οι ρυθμίσεις που απαιτούνται στο STM32CubeIDE. Αφού οριστούν αυτές οι ρυθμίσεις, ο IDE θα δημιουργήσει αυτόματα τον κώδικα που απαιτείται για να δουλέψει το παράδειγμα.

▼ STMicroelectronics.X-CUBE-BLE1	✓	6.2.4	▼	
> Exposed APIs	✓			
▼ Wireless BlueNRG-MS	✓	5.1.3		
▼ BlueNRG-MS	✓			
Controller	✓	5.1.3	<input checked="" type="checkbox"/>	
HCI_TL	✓	5.1.3	Basic	▼
HCI_TL_INTERFACE	✓	5.1.3	UserBoard	▼
Utils	✓	5.1.3	<input checked="" type="checkbox"/>	
▼ Device BLE1_Applications	✓	6.1.4		
Application	✓	6.1.4	SampleApp	▼

Σχήμα B.1: X-CUBE-BLE1 SampleApp Init



UFBGA132 (Top view)

Σχήμα B.2: X-CUBE-BLE1 Pinout Configuration

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

Configure the below parameters :

Search (Ctrl+F)

Basic Parameters

Frame Format Motorola

Data Size 8 Bits

First Bit MSB First

Clock Parameters

Prescaler (for Baud Rate) 16

Baud Rate 250.0 KBits/s

Clock Polarity (CPOL) Low

Clock Phase (CPHA) 1 Edge

Advanced Parameters

CRC Calculation Disabled

NSSP Mode Enabled

NSS Signal Type Software

Σχήμα B.3: X-CUBE-BLE1 SPI1 Configuration

Configuration

NVIC Code generation

Priority Group 3 bits for pre-... Sort by Preemption Priority and Sub Priority Sort by interrupts names

Search Search... Show available interrupts Force DMA channels interrupts

NVIC_NS Interrupt Table	Enabled	Preemption Priority	Sub Priority
Memory management fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	7	0
PVD/PVM1/PVM2/PVM3/PVM4 interrupts through EXTI lines 16/35/36/37/38	<input type="checkbox"/>	0	0
Flash non-secure global interrupt	<input type="checkbox"/>	0	0
RCC non-secure global interrupt	<input type="checkbox"/>	0	0
EXTI line6 interrupt	<input checked="" type="checkbox"/>	0	0
EXTI line13 interrupt	<input checked="" type="checkbox"/>	0	0
SPI1 global interrupt	<input type="checkbox"/>	0	0
USART1 global interrupt / USART1 wake-up interrupt through EXTI line 26	<input type="checkbox"/>	0	0
FPU global interrupt	<input type="checkbox"/>	0	0

Σχήμα B.4: X-CUBE-BLE1 NVIC Configuration

STMicroelectronics X-CUBE-BLE1.6.2.4_M33NS Mode and Configuration

Mode

Runtime contexts:

Cortex-M33 secure	Cortex-M33 non secure
<input type="radio"/>	<input checked="" type="radio"/>

☒ Wireless BlueNRG-MS

☒ Device BLE1 Applications

Configuration

Reset Configuration

Parameter Settings User Constants Platform Settings

Platform proposal

HCI_TL_INTERFACE

Name	IPs or Components	Found Solutions	BSP API
Exti Line	GPIO:EXTI	PG6	HAL_EXTI_DRIVER
BUS IO driver	SPI:Full-Duplex Master	SPI1	BSP_BUS_DRIVER
CS Line	GPIO:Output	PG5	Unknown
Reset Line	GPIO:Output	PG8	Unknown

BSP

Name	IPs or Components	Found Solutions	BSP API
BSP BUTTON	GPIO:EXTI	PC13	BSP_COMMON_DRIVER
BSP USART	USART:Asynchronous	USART1	BSP_COMMON_DRIVER
BSP LED	GPIO:Output	PG12	BSP_COMMON_DRIVER

Σχήμα B.5: X-CUBE-BLE1 Mode and Configuration

Το SampleApp χρησιμεύει ως σημείο εκκίνησης για προγραμματιστές που θέλουν να δημιουργήσουν τις δικές τους εφαρμογές BLE καθώς δείχνει τα ακόλουθα χαρακτηριστικά του πακέτου X-CUBE-BLE1:

- Advertising και scanning για BLE συσκευές
- Δημιουργία BLE σύνδεσης μεταξύ κεντρικής και περιφερειακής συσκευής
- Ανάγνωση και εγγραφή δεδομένων μέσω σύνδεσης BLE
- Ρύθμιση custom GATT (Generic Attribute) service με custom characteristics

ΠΑΡΑΡΤΗΜΑ Γ

TrustZone

Γ.1 Απενεργοποίηση του TrustZone

Μόλις ενεργοποιηθεί το TrustZone στη συσκευή, μπορεί να απενεργοποιηθεί μόνο κατά τη διάρκεια μιας "υποβάθμισης" RDP στο επίπεδο 0 (είτε από το επίπεδο RDP 1 στο επίπεδο 0 είτε από το επίπεδο RDP 0,5 στο επίπεδο 0). Έτσι, η απενεργοποίηση του TrustZone οδηγεί σε πλήρη διαγραφή του τσιπ. Η απενεργοποίηση του TrustZone μπορεί να γίνει μόνο μέσω του Bootloader ή της διεπαφής εντοπισμού σφαλμάτων (JTAG/SWD). Αυτό εγγυάται ότι ένα κακόβουλο λογισμικό δεν μπορεί να απενεργοποιήσει το TrustZone.

Σημείωση: Εάν η συσκευή βρίσκεται σε επίπεδο RDP 2, όλες οι δυνατότητες εντοπισμού σφαλμάτων είναι απενεργοποιημένες και η εκκίνηση από τη μνήμη συστήματος (λειτουργία εκκίνησης) δεν είναι πλέον διαθέσιμη. Κατά συνέπεια, στο επίπεδο RDP 2, είναι αδύνατο να απενεργοποιηθεί το TrustZone.

Σε επίπεδο RDP 0,5 και επίπεδο 1:

- Όταν η CPU βρίσκεται σε ασφαλή κατάσταση, δεν είναι δυνατή η σύνδεση με το MCU μέσω JTAG/SWD, επομένως δεν είναι δυνατή η υποβάθμιση TZEN/RDP.
- Όταν η CPU είναι σε μη ασφαλή κατάσταση, είναι δυνατή η σύνδεση με το MCU μέσω JTAG/SWD και υποβάθμισης RDP.

Για εκκίνηση από τη μνήμη Flash, εάν ο μη ασφαλής κωδικός δεν καλείται από τον ασφαλή κωδικό, η CPU παραμένει πάντα σε ασφαλή κατάσταση και η υποβάθμιση RDP δεν μπορεί να γίνει μέσω JTAG / SWD. Πριν προγραμματιστεί το επίπεδο RDP 0,5 ή το επίπεδο RDP 1, ο χρήστης πρέπει πάντα να διασφαλίζει ότι η ασφαλής εφαρμογή καλεί τη μη ασφαλή εφαρμογή, έτσι ώστε να είναι δυνατή η σύνδεση με τον στόχο.

- Βήμα 1: Flush το παράδειγμα "Blinky" 3.4.1.

- Βήμα 2: Ρυθμίστε το επίπεδο RDP από τα option bytes στο επίπεδο 1.

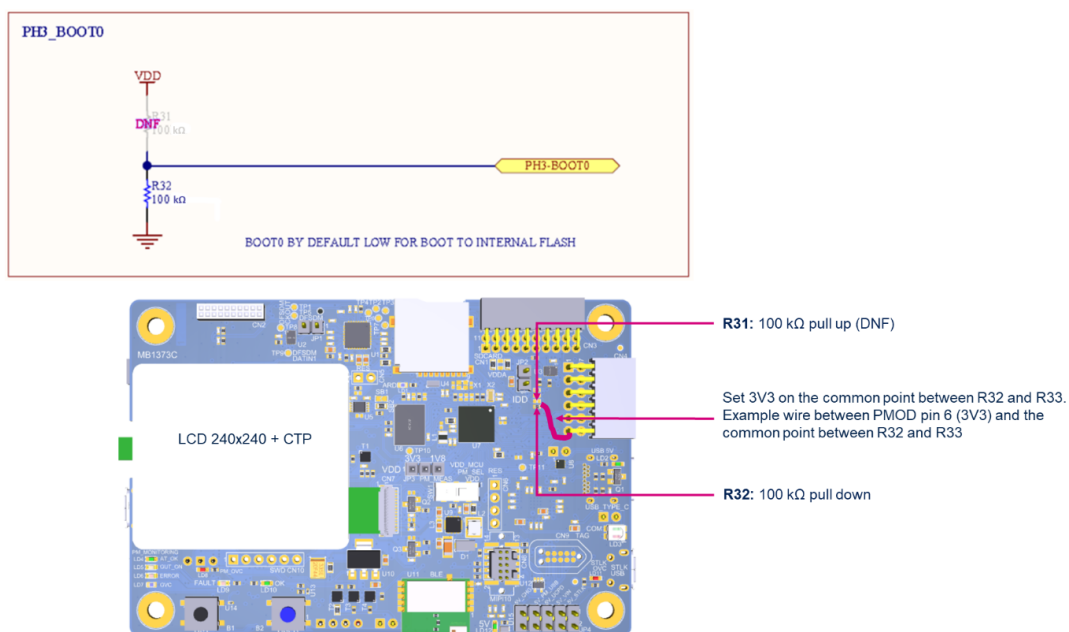
Σημείωση: Όταν το επίπεδο RDP ρυθμιστεί στο επίπεδο 1, πρέπει να χρησιμοποιηθεί διαφορετικό τροφοδοτικό από το USB ST-LINK για σύνδεση στο MCU.

- Βήμα 3: Ρυθμίστε το option byte επίπεδο RDP και TZEN στο 0.

Εάν το βήμα 1 παραλείφθηκε, ο μόνος τρόπος να γίνει υποβάθμιση RPD είναι μέσω boot-loader με εκκίνηση από RSS. (Το RSS είναι ενσωματωμένο στο μπλοκ ασφαλών πληροφοριών, μέρος της περιοχής ασφαλούς μνήμης Flash και προγραμματίζεται κατά την παραγωγή του ST MCU)[39]

Σημείωση: Αυτό συμβαίνει επειδή υπάρχει ένα άλμα από το RSS (ασφαλές) στο Bootloader (μη ασφαλές), επομένως με μια εκκίνηση από RSS η κατάσταση της CPU είναι εγγυημένη ότι θα αλλάξει από ασφαλή σε μη ασφαλή και είναι δυνατή η σύνδεση με τον στόχο.

Στο STM32L562E-DK, το PH3-BOOT0 είναι σταθερό σε LOW επιτρέποντας την εκκίνηση από τη διεύθυνση μνήμης που ορίζεται από το option byte SECBOOTADD0. Για να γίνει εκκίνηση από RSS, χρειάζεται το PH3-BOOT0 να τεθεί σε HIGH εφαρμόζοντας 3V3 στο σήμα PH3-BOOT0 μεταξύ R32 και R31.[40]



Σχήμα Γ.1: Εκκίνηση από RSS (Τροποποίηση BOOT0)

Γ.2 Αλλαγή Option Bytes runtime

To struct FLASH_OBProgramInitTypeDef περιέχει όλες τις απαραίτητες πληροφορίες για τον προγραμματισμό και την ανάγνωση των option bytes. Χρησιμοποιώντας το OB.USERConfig, μπορούμε να ελέγξουμε την τιμή του καταχωρητή OPTR, ο οποίος περιέχει την τρέχουσα τιμή των option byte του χρήστη. Πάντα πρέπει να ελέγχουμε εάν τα option byte είναι ήδη στην επιθυμητή διαμόρφωση πριν ξεκλειδώσουμε άσκοπα τη μνήμη flash για να τα αλλάξουμε. Ένα άλλο πράγμα που πρέπει να σημειωθεί είναι ότι η HAL_FLASH_OB_Launch() προκαλεί επαναφορά συστήματος και επομένως δεν θα επιστρέψει ποτέ. Αυτός είναι ο λόγος για τον οποίο επιστρέφεται ένα HAL_ERROR μετά τη λειτουργία εκκίνησης, καθώς δεν πρέπει ποτέ να επιτευχθεί.

Ας πούμε, για παράδειγμα, ότι θέλουμε να τροποποιήσουμε τα byte nRST_STOP, nRST_STDBY και nRST_SHDW ώστε να είναι 1, 0 και 1 αντίστοιχα:

```
1  HAL_StatusTypeDef modifynRST()
2  {
3      FLASH_OBProgramInitTypeDef OB;
4      HAL_FLASHEx_OBGetConfig(&OB);
5
6                      // check if
7      if ( !(OB.USERConfig & FLASH_OPTR_nRST_STOP) || // nRST_STOP is cleared
           or
8          (OB.USERConfig & FLASH_OPTR_nRST_STDBY) || // nRST_STDBY is set      or
9          !(OB.USERConfig & FLASH_OPTR_nRST_SHDW) ) // nRST_SHDW is cleared
10     {
11
12         HAL_FLASH_Unlock();
13         HAL_FLASH_OB_Unlock();
14
15         OB.OptionType = OPTIONBYTE_USER;
16         OB.USERType = OB_USER_nRST_STOP | OB_USER_nRST_STDBY | OB_USER_nRST_SHDW;
17         OB.USERConfig = OB_STOP_NORST | OB_STANDBY_RST | OB_SHUTDOWN_NORST;
18
19         if ( HAL_FLASHEx_OBProgram(&OB) != HAL_OK )
20         {
21             HAL_FLASH_OB_Lock();
22             HAL_FLASH_Lock();
23             return HAL_ERROR;
24         }
25     }
```

```

26     HAL_FLASH_OB_Launch();
27
28     /* We should not make it past the Launch, so lock
29      * flash memory and return an error from function
30      */
31     HAL_FLASH_OB_Lock();
32     HAL_FLASH_Lock();
33     return HAL_ERROR;
34 }
35
36 return HAL_OK;
37 }

```

Αρχικά, ελέγχουμε εάν κάποια από τις επιθυμητές επιλογές έχει ρυθμιστεί ήδη. Στη συνέχεια, ορίζουμε στο OB.USERType τον συνδυασμό των επιλογών που θέλουμε να προγραμματίσουμε και στο OB.USERConfig τον συνδυασμό τιμών για αυτές τις επιλογές.[41]