

**Kajian Penggunaan *Secure Microcontroller*
sebagai Solusi Pengembangan
Sistem Embedded yang Aman**

disusun sebagai

**Tugas Akhir Kuliah
EC5010 Keamanan Sistem Informasi**



ditulis oleh:

Yusdi Saliman

13203047

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

2006

Kajian Penggunaan *Secure Microcontroller* sebagai Solusi Pengembangan Sistem Embedded yang Aman

Abstraksi

Keamanan sistem informasi umumnya lebih banyak mengarah ke sistem yang melibatkan komputer PC (*Personal Computer*). Pada kenyataannya, sistem informasi juga melibatkan banyak sistem-sistem lain baik yang lebih besar dan kompleks maupun yang lebih kecil dan sederhana seperti *embedded system* (sistem embedded). Umumnya dalam desain embedded system, aspek security adalah aspek yang seringkali dilupakan, atau setidaknya bukan prioritas utama. Hal ini bukannya tanpa alasan. Bagaimana pun juga embedded system adalah sistem yang kecil dengan resource dan processing power yang sangat terbatas, sedangkan menambahkan implementasi security (misal kriptografi) berarti memberikan tambahan kebutuhan akan dua hal tersebut, yang akhirnya mungkin berakibat pada sistem yang menjadi semakin kompleks, dan lebih buruk lagi, semakin mahal. Namun demikian, dengan semakin kritisnya aplikasi-aplikasi yang dialihkan ke embedded system, apalagi semakin banyak embedded system yang terhubung dengan internet, kebutuhan security dalam embedded system semakin terasa.

Ada sejumlah aspek yang membuat embedded system berbeda dari sistem yang “biasa”, dan beberapa masalah yang mungkin muncul juga berbeda sehingga perlu penanganan yang lebih khusus. *Secure microcontroller* merupakan salah satu solusi yang ditawarkan untuk embedded system security, yaitu mikrokontroler yang program dan datanya terlindungi dengan enkripsi ditambah deteksi terhadap berbagai serangan, serta biasanya juga menawarkan dukungan kriptografi secara hardware kepada programmernya dengan *cryptographic engine*. Contohnya adalah DS5250 buatan Dallas Semiconductor, sebuah secure microcontroller yang kompatibel dengan Intel 8051, dan keluarga AT90SC buatan Atmel, yaitu yang kompatibel dengan Atmel AVR.

I. Security di dalam Embedded System

Dengan semakin luasnya penggunaan embedded system, pentingnya aspek security menjadi semakin terasa. Terhubungnya embedded system ke jaringan yang luas seperti internet selain memudahkan penggunaanya juga sekaligus membuka peluang terjadinya serangan oleh pihak lain.

Sebelum melihat masalah-masalah security apa saja yang mungkin muncul dalam embedded system, pertama-tama akan dilihat aspek-aspek apa saja yang membedakan embedded system dengan sistem komputer lainnya, tentu saja yang berhubungan dengan security di embedded system.

Perlu diperhatikan bahwa pembahasan di sini hanya memfokuskan pada aspek security dari embedded system, dan mengasumsikan bahwa pembaca sudah cukup familiar dengan istilah-istilah yang berhubungan dengan embedded system dan arsitektur komputer serta istilah yang umum dalam information security.

1.1. Aspek-aspek yang membedakan Embedded System dari sistem lain

a. Biaya (cost)

Ini merupakan aspek yang dapat dikatakan paling penting karena sangat mempengaruhi desain suatu embedded system secara keseluruhan. Dalam membuat suatu embedded system, biasanya dipilih komponen-komponen secara optimal, yaitu yang memungkinkan implementasi sistem tersebut tetapi dengan biaya yang serendah-rendahnya. Hal ini karena perbedaan harga sedikit saja dapat sangat berpengaruh ketika embedded system tersebut harus dipasarkan secara luas dalam jumlah yang besar.

Cost ini akhirnya berkaitan erat juga dengan pengertian cost dalam hardware komputer dan elektronika. Untuk menekan biaya tersebut, caranya adalah dengan menggunakan resource komputer yang sesedikit mungkin, komponen elektronik seperlunya, dan sebagainya. Akibatnya adalah seperti yang kita ketahui, embedded system merupakan sistem dengan kemampuan komputasi dan ketersediaan resource yang relatif rendah dibandingkan sistem komputer yang lebih besar misalnya komputer desktop.

Keterbatasan ini akhirnya secara langsung berpengaruh juga pada security. Penambahan mekanisme otentikasi, pengamanan data dengan kriptografi, dan sebagainya berarti juga penambahan kebutuhan akan resource tersebut. Dengan sistem yang sudah ada bisa jadi resource untuk kebutuhan tersebut tidak tersedia.

b. Constraint waktu

Tidak sedikit embedded system yang sekaligus merupakan *real-time system* (sistem nyata-waktu), yaitu sistem yang prosesnya terbatas oleh batas waktu. Sistem-sistem ini umumnya merupakan sistem yang digunakan untuk keperluan yang kritis, dan harus selalu aktif. Dengan demikian tidak seperti sistem komputer desktop yang dapat dilakukan *reboot*, misalnya untuk menjaga kestabilannya atau menangani serangan tertentu seperti virus, dalam embedded system tertentu hal tersebut mungkin tidak dapat diterima. Embedded system harus selalu stabil, termasuk dalam gangguan oleh serangan. Harus diperhatikan bagaimana jika suatu real-time system mengalami serangan *Denial of Service* (DoS) yang membuatnya menjadi lambat sehingga batas waktunya tidak lagi terpenuhi.

c. Interaksi langsung dengan dunia nyata

Banyak embedded system, umumnya embedded control application, harus berhubungan langsung dengan dunia nyata. Akibatnya adalah kesalahan akibat suatu gangguan bisa berakibat lebih fatal dibandingkan sistem komputer yang biasa. Jika misalnya suatu komputer server yang menyimpan database mengalami gangguan, paling parah yang terjadi adalah kehilangan data, dan apabila database tersebut di-backup secara berkala maka kerugiannya lebih kecil lagi. Hal ini akan sangat berbeda jika misalnya sistem kontrol dalam suatu pabrik kimia mengalami gangguan dan melakukan kesalahan.

d. Constraint energi

Banyak embedded system yang mengambil daya dari baterai. Hal ini berarti munculnya satu titik serangan baru pada embedded system, yaitu power supply.

e. Elektronika

Masih berhubungan dengan yang terakhir, karena embedded system merupakan sistem yang sangat erat dengan elektronika, maka serangan-serangan atau gangguan juga mungkin dilakukan secara elektrik, misalnya analisis dengan multimeter, logic analyzer, dan sebagainya. Walaupun sistem komputer lain pada dasarnya juga merupakan alat elektronik, tetapi kemungkinan hal ini dilakukan lebih tinggi untuk embedded system.

1.2. Masalah-masalah Security pada Embedded System

a. Masalah Kecepatan dan Keterbatasan Resource

Hal ini sudah disinggung sebelumnya, yaitu bahwa kemampuan suatu embedded system terbatas oleh kemampuan pemrosesan dan resource yang ada. Hal ini menjadi masalah ketika diperlukan hal-hal misalnya implementasi public-key cryptography. Proses perhitungannya selain memakan memori juga membutuhkan kemampuan proses yang tinggi. Jika misalnya embedded system dikembangkan menggunakan mikrokontroler dengan kecepatan clock hanya 12 MHz dan memori data sebesar 256 byte kemungkinan besar proses ini akan memakan waktu yang lama. Hal ini bisa mengakibatkan ketidaknyamanan, dan pada kasus real-time system, sama sekali tidak dapat diterima.

b. Physical Access (tampering)

Seperti pada sistem komputer yang biasa, embedded system juga rentan terhadap serangan di mana si penyerang dapat mengakses langsung sistem tersebut secara fisik. Walaupun masalah ini juga ada pada sistem komputer yang biasa, tetapi yang dilakukan terhadap embedded system berbeda dari yang dilakukan terhadap komputer PC. Pada komputer PC, mungkin penyerang akan melakukan booting dengan sebuah rescue disk dan mencuri file yang berisi password untuk di-crack nantinya, tetapi penyerang umumnya tidak akan sampai melihat rangkaian motherboard dari PC yang diserangnya. Berbeda dengan cara tersebut, penyerang pada embedded system justru umumnya akan membuka kemasan alat dan melihat sampai ke dalamnya dan melakukan analisis secara

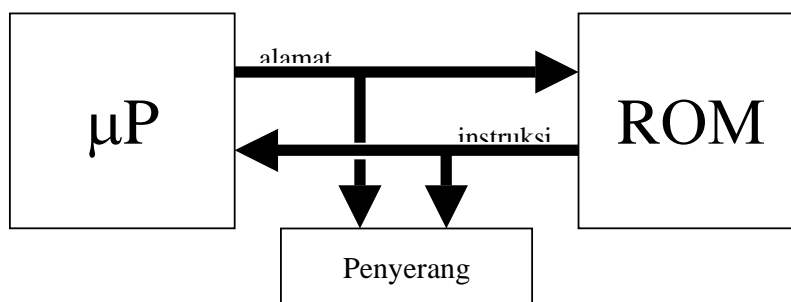
elektrik. Tindakan ini disebut *tampering*. Umumnya serangan ini adalah untuk melakukan *reverse engineering*.

Yang dilakukan bisa bermacam-macam. Penyerang bisa melihat rangkaian dari sistem dan membuat skematiknya. Hal ini bisa bertujuan untuk membuat tiruan dari sistem tersebut, atau untuk dianalisis lebih lanjut untuk menentukan titik yang paling mudah untuk diserang. Penyerang juga mungkin mengambil mikrokontroler atau chip memori yang berisi program yang dijalankan. Walaupun kebanyakan memori program dilindungi dengan security lock bits, tetapi sudah banyak yang mampu mengalahkan sistem keamanan tersebut. Dengan membaca isi memori tersebut, firmware (program yang dijalankan) dapat diketahui. Sama seperti sebelumnya, ini bisa bertujuan untuk meniru atau untuk analisis lebih lanjut.

c. Serangan secara digital

Masih berhubungan dengan physical access, analisis lebih lanjut masih bisa dilakukan apabila penyerang dapat berinteraksi langsung dengan rangkaiannya. Dengan menggunakan multimeter, logic probe, logic analyzer, atau alat-alat elektronik lainnya, penyerang dapat menganalisis nilai-nilai yang ada pada suatu pin IC tertentu, nilai yang ada di bus data dan alamat, dan sebagainya.

Misalnya penyerang tidak berhasil mendapatkan firmware dengan melihat isi memori program, penyerang masih mungkin mendapatkan firmware tersebut apabila ia menganalisis nilai-nilai yang ada pada bus data dan alamat yang digunakan ketika prosesor melakukan *instruction fetch*. Ilustrasinya seperti berikut ini:



Lebih jauh lagi, apabila program telah diketahui, penyerang mungkin menganalisis data di RAM dengan cara yang sama. Misalnya program menggunakan mekanisme kriptografi untuk mengamankan data yang dikirim melalui internet, apabila program telah diketahui, penyerang mungkin dapat mengamati bus alamat dan data yang terhubung ke RAM untuk mendapatkan kunci dari enkripsi yang digunakan. Atau jika penyerangnya cukup pintar, ia mungkin saja dapat memberi instruksi palsu di memori program yang menyuruh prosesor untuk mengoutput kunci tersebut melalui I/O port.

Apabila embedded system tersebut menggunakan PLD (*Programmable Logic Device*) atau FPGA (*Field Programmable Gate Array*), penyerang mungkin saja melakukan *I/O scan* untuk melihat karakteristik input/output dari rangkaian digital yang diimplementasikan di dalamnya.

d. Serangan secara elektrik

Karena embedded system merupakan sistem elektronika, maka serangan secara elektrik juga dimungkinkan.

Serangan terhadap power supply

Yang paling sederhana adalah dengan menyerang power supply, seperti yang sudah disinggung sebelumnya, karena banyak embedded system yang mengambil daya dari baterai. Serangan paling mudah adalah dengan menghilangkan supply daya tersebut. Hal ini jelas dapat dilakukan dengan mudah apabila penyerang dapat secara fisik mengakses alat yang diserang. Namun demikian perlu diperhatikan juga bahwa hal ini juga dapat dilakukan secara *remote*.

Misalnya, suatu embedded system terhubung dengan internet dan akan melakukan sesuatu apabila penggunanya memberikan suatu perintah tertentu. Untuk menghemat energi yang tersedia, mikroprosesor akan masuk ke keadaan *idle mode* sampai ia menerima perintah. Serangan dapat dilakukan dengan terus-menerus mengirimkan data ke embedded system tersebut melalui internet. Sekalipun misalnya ada mekanisme authentication dan authorization untuk

menjamin bahwa perintah memang dikirimkan oleh pengguna yang seharusnya, setiap data yang diterima tetap harus diproses, misalnya untuk menentukan apakah pengirimnya memang orang yang benar. Dengan terus-menerus mengirimkan data ke embedded system, prosesor menjadi bekerja terus dan konsumsi dayanya menjadi tinggi. Hal ini bisa berakibat pada habisnya energi baterai sebelum waktu yang diperkirakan.

Serangan dengan memanfaatkan power supply bisa lebih bervariasi apabila penyerangnya memiliki akses secara fisik. Misalnya, dengan mengubah besarnya tegangan V_{CC} , mungkin bisa terjadi hal-hal yang tidak diinginkan, seperti meng-clear-kan security lock bits yang melindungi firmware dalam mikrokontroler (menurut literatur hal ini dapat dilakukan terhadap mikrokontroler PIC16C84).

Inferensi Elektromagnetik (EMI)

Karena embedded system merupakan rangkaian listrik, maka ketika beroperasi ia akan menghasilkan emisi berupa inferensi elektromagnetik (*electromagnetic inference*) atau EMI. Serangan dengan memanfaatkan ini bisa secara aktif maupun pasif. Secara aktif, embedded system diserang dengan misalnya membuat sinyal radio dengan frekuensi tinggi yang akan menghasilkan EMI yang mampu merusak atau mengganggu kerja sistem. Secara pasif serangan dapat dilakukan dengan menangkap EMI yang dihasilkan sistem dan menganalisisnya.

Timing Attack

Serangan juga dapat dilakukan dengan memanfaatkan waktu operasi sistem, baik secara aktif maupun pasif. Secara aktif hal ini dilakukan dengan mengubah kecepatan clock dari rangkaian, tentu saja ini membutuhkan physical access. Tujuan dari hal ini bisa untuk menyebabkan sistem melakukan pekerjaannya dengan tidak benar (hal ini bisa fatal jika dilakukan terhadap real-time system), atau untuk memudahkan proses analisis secara digital yang disebutkan sebelumnya dengan memperlambat clock. Serangan secara pasif dilakukan dengan mengukur waktu untuk menentukan mekanisme kerja sistem.

e. Serangan melalui *external interface*

External interface juga dapat digunakan penyerang untuk menganalisis sistem yang diserangnya. Misalnya jika sistem tersebut berkomunikasi dengan devais lain melalui protokol RS-232, penyerang dapat saja melakukan *eavesdropping* terhadap jalur komunikasi tersebut untuk melakukan analisis. Setelah komunikasi yang dilakukan sudah dimengerti, penyerang bisa saja mengirimkan data-data tertentu melalui interface tersebut untuk merusak atau mengganggu sistem.

Masalah dengan external interface juga mungkin terjadi tanpa harus ada komunikasi dengan devais lain. Misalnya, pada board rangkaian sistem yang dibuat terdapat socket JTAG yang digunakan selama proses debugging. Jika penyerangnya cukup pintar, ia dapat memanfaatkan interface tersebut untuk melakukan berbagai hal, misalnya saja mengubah program.

f. Firmware upgrade

Saat ini banyak embedded system yang menawarkan fasilitas untuk mengupgrade firmware. Di satu sisi hal ini selain membuat penggunaanya lebih nyaman juga bisa meningkatkan security, karena perbaikan terhadap security hole dapat dimungkinkan pada firmware yang lebih baru. Namun di sisi lain ini bisa menjadi masalah security yang baru. Apabila firmware dapat diupgrade, maka terbuka juga peluang bahwa firmware tersebut diupgrade oleh pihak yang salah, misalnya virus. Selain itu, hal ini juga berarti bahwa firmware yang baru dapat tersedia di luar sistem tersebut, misalnya dapat didownload dari internet. Artinya, tanpa berhubungan langsung dengan embedded system yang akan diserang, seorang penyerang dapat saja melakukan reverse engineering terhadap firmware yang baru tersebut.

g. Masalah-masalah yang juga ada pada sistem komputer lain

Masalah-masalah yang dibahas di atas adalah masalah-masalah yang spesifik pada embedded system. Selain masalah-masalah tersebut, berbagai masalah security klasik yang terjadi pada sistem komputer lain mungkin juga

terjadi pada embedded system sehingga perlu dipikirkan juga. Masalah ini misalnya masalah yang berhubungan dengan pemrograman seperti buffer overflow, serangan yang berhubungan dengan jaringan seperti traffic analysis, dan juga serangan yang tidak bersifat teknis tetapi bisa dianggap paling efektif, yaitu social engineering.

II. Secure Microcontroller

Sebagai solusi untuk masalah-masalah security dalam embedded system seperti yang sudah dibahas di atas, beberapa perusahaan pembuat mikrokontroler membuat jenis mikrokontroler khusus yang disebut *secure microcontroller*. Secure microcontroller adalah mikrokontroler yang program dan datanya dapat dilindungi dengan enkripsi, dan biasanya juga menyediakan fasilitas untuk mempercepat proses kriptografi. Selain itu biasanya diberikan juga pengamanan-pengamanan lain misalnya pengamanan terhadap tampering.

Sebagai contoh secure microcontroller, di sini akan digunakan DS5250 buatan Dallas Semiconductor dan keluarga AT90SC buatan Atmel. DS5250 merupakan mikrokontroler yang kompatibel dengan mikrokontroler Intel 8051, sedangkan AT90SC kompatibel dengan mikrokontroler Atmel AVR. Pembahasan tentang keduanya di sini dikhususkan pada aspek embedded security yang ditawarkan, bukan pada arsitekturnya atau cara pemakaiannya.

2.1. Security features yang dimiliki DS5250 dan AT90SC

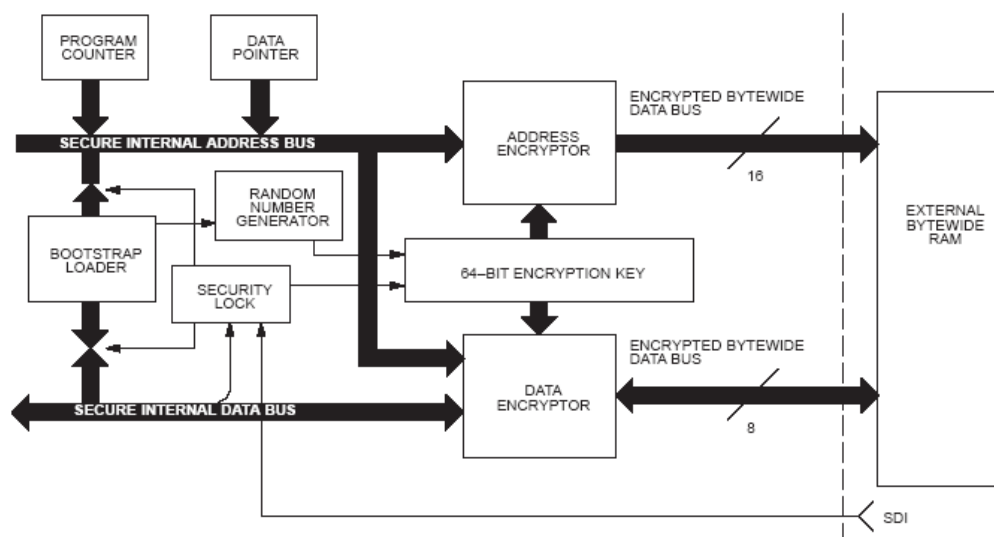
a. Non-volatile memory dengan ukuran yang besar

DS5250 menggunakan teknologi NV RAM (Non Volatile RAM) sebagai memori utamanya, dengan ukuran sebesar 4 MB. AT90SC tidak menggunakan NV RAM sebagai memori datanya, tetapi selain SRAM AT90SC juga menyediakan EEPROM dengan ukuran yang besar yang dapat digunakan untuk penyimpanan data secara non-volatile. Pada AT90SC12836RCFT ukuran EEPROM tersebut adalah 36 KB. Walaupun tidak sebesar pada DS5250 tetapi untuk kebanyakan embedded system ukuran ini juga relatif besar.

Non-volatile memory memang tidak secara langsung berhubungan dengan aspek security, tetapi dapat membantu untuk mengimplementasikan pengamanan data. Memori non-volatile penting misalnya sebagai media untuk penyimpanan kunci yang digunakan dalam kriptografi. Kunci ini harus disimpan dalam memori yang non-volatile karena jika tidak maka kunci akan hilang apabila sistem dimatikan. Dengan ukuran non-volatile memory yang besar banyak data yang bisa diamankan, sehingga apabila daya hilang maka data-data tersebut tidak ikut hilang.

b. Enkripsi terhadap memori program dan data

Baik DS5250 maupun AT90SC mendukung dilakukannya enkripsi pada memori program dan data. Enkripsi ini berlaku baik untuk data yang dibaca dan ditulis maupun untuk alamat. Hal ini semakin penting pada DS5250 yang mendukung pengaksesan ke memori eksternal. Ketika mikrokontroler akan mengakses memori eksternal, baik data maupun alamat yang melalui bus adalah nilai yang telah terenkripsi. Pada DS5250 jenis enkripsi yang digunakan adalah DES atau triple-DES. Ilustrasinya diperlihatkan oleh blok diagram dari enkripsi software pada DS5250:



c. Cryptographic engine

DS5250 memiliki *DES engine* yang dapat melakukan enkripsi dan dekripsi DES dan triple-DES. Fasilitas ini dapat dimanfaatkan programmer untuk melakukan proses enkripsi dan dekripsi DES secara lebih mudah dan lebih cepat. Selain itu, DS5250 juga memiliki *modulo arithmetic accelerator* (MAA). MAA ini berguna untuk proses kriptografi kunci publik RSA yang memerlukan operasi yang melibatkan modulus dengan bilangan yang besar. Selain itu juga terdapat *random number generator* dan dukungan CRC (*cyclic redundancy check*).

Sama seperti DS5250, AT90SC juga memberi dukungan hardware yang serupa (bahkan lebih lengkap) untuk kriptografi. AT90SC juga memiliki DES engine sama seperti DS5250, ditambah sebuah *cryptographic accelerator* yang bernama AdvX. Kriptografi yang didukung oleh AdvX adalah RSA, DSA (*Digital Signature Algorithm*), DH (*Diffie-Hellman*), dan ECC (*Eliptic Curve Cryptography*). Sama seperti DS5250, AT90SC juga memiliki random number generator dan mendukung CRC, ditambah *checksum accelerator*.

d. Watchdog timer

Fasilitas ini sebetulnya sudah umum dan tidak terbatas hanya pada secure microcontroller, tetapi karena *watchdog timer* juga dapat dimanfaatkan untuk pengamanan maka dimasukkan juga di sini. Watchdog timer adalah fitur berupa timer yang digunakan untuk menjamin program berjalan secara normal. Watchdog timer dapat dibuat untuk mulai menghitung, dan setelah hitungan sampai ke suatu nilai tertentu, mikrokontroler otomatis akan melakukan reset. Untuk mencegah reset terjadi, pada program diberikan instruksi untuk me-reset hitungan watchdog timer secara berkala. Apabila karena suatu hal program berjalan tidak sesuai alur yang semestinya, instruksi tersebut kemungkinan besar tidak tereksekusi sehingga watchdog timer akan menghitung sampai nilai batasnya dan mikrokontroler akan melakukan reset. Baik DS5250 maupun AT90SC mendukung penggunaan watchdog timer.

e. Proteksi terhadap gangguan fisik dan elektrik

DS5250 dan AT90SC dilengkapi berbagai sensor internal yang dapat menangkap perubahan pada lingkungan yang menandakan adanya gangguan fisik. Sensor-sensor tersebut di antaranya adalah sensor tegangan dan temperatur. Pada AT90SC juga terdapat sensor frekuensi. DS5250 dan AT90SC juga dapat mendeteksi adanya tampering dan *probing*. Proteksi terhadap probing dilakukan dengan menambahkan lapisan metal khusus. Pada DS5250, jika probing dilakukan maka lapisan metal ini akan menyebabkan terjadinya hubung singkat pada bagian tertentu rangkaian di dalam mikrokontroler yang mengakibatkan kerusakan data. Jadi apabila tampering atau probing dilakukan terhadap mikrokontroler, mikrokontroler secara otomatis akan menghancurkan data-data yang ada.

2.2. Security features yang dimiliki DS5250 tetapi tidak dimiliki AT90SC

a. Vector RAM

Dalam keadaan memori program terenkripsi, DS5250 juga memanfaatkan fasilitas *Vector RAM*, yaitu RAM khusus yang memetakan ke alamat 48 byte pertama dari memori program. Tujuannya adalah untuk mengamankan *reset vector* dan *interrupt vector*. Tanpa vector RAM, penyerang masih mungkin menganalisis program dengan memaksa dilakukannya reset atau interrupt tertentu, kemudian melihat alamat terenkripsi yang dihasilkan pada bus alamat eksternal. Dengan mengetahui alamat seharusnya dan alamat terenkripsi ini penyerang dapat menentukan kunci enkripsinya. Dengan adanya vector RAM masalah ini dapat diatasi, karena alamat vektor yang dienkripsi bukan alamat vektor yang biasanya digunakan pada mikrokontroler Intel 8051 yang standard.

Fitur ini tidak terdapat pada AT90SC, tetapi dapat dikatakan bahwa fitur ini kurang diperlukan pada AT90SC, karena AT90SC memang tidak mendukung memori program eksternal. Dengan kata lain teknik analisis seperti di atas hanya dapat dilakukan apabila penyerang bisa mengakses bus alamat internal mikrokontroler. Asalkan mikrokontroler dilengkapi dengan proteksi tampering yang baik maka peluang analisis alamat vektor seperti di atas tidak akan ada.

b. Dummy Bus Access

Sebagai tambahan pengamanan untuk mencegah berhasilnya analisis terhadap bus eksternal, DS5250 juga menambahkan *dummy bus access*. Dummy bus access adalah penggunaan bus oleh mikrokontroler yang sesungguhnya tidak diperlukan. Jadi mikrokontroler memberi alamat ke bus dan membaca data dari memori, tetapi kemudian data yang dibaca dibuang setelah sampai di mikrokontroler. Dengan memasukkan banyak dummy bus access di antara akses-akses yang sesungguhnya, analisis terhadap bus eksternal menjadi lebih sulit, atau bahkan tidak mungkin dilakukan.

Sama seperti vector RAM, AT90SC tidak membutuhkan teknik ini karena memang tidak mendukung pengaksesan memori eksternal.

c. Self-Destruct input

Walaupun proteksi terhadap gangguan fisik terhadap mikrokontroler dapat mengatasi tampering terhadap mikrokontroler, proteksi tersebut belum melindungi sistem secara keseluruhan dari tampering ke alat elektronik yang dibuat. DS5250 memiliki 2 pin khusus yaitu *self-destruct input* (SDI). Pin yang pertama akan menyebabkan hilangnya data-data di memori apabila aktif, sedangkan pin yang kedua akan memberikan interrupt apabila aktif, jadi programmer juga bisa memberikan respons tertentu secara software selain self-destruct.

AT90SC tidak memiliki fitur ini, padahal ini merupakan fitur yang bisa sangat berguna untuk mengamankan sistem dari tampering.

2.2. Security features yang dimiliki AT90SC tetapi tidak dimiliki DS5250

a. Dukungan kriptografi yang lebih lengkap

Hal ini sudah dibahas sebelumnya pada bagian cryptographic engine. Walaupun DS5250 dan AT90SC sama-sama memberikan dukungan hardware untuk kriptografi, dukungan yang diberikan AT90SC lebih lengkap, karena selain DES dan RSA yang juga didukung DS5250, AT90SC juga mendukung DSA, DH, dan ECC.

b. Internal Oscillator

Fitur ini sebetulnya tidak disebutkan sebagai fitur security oleh Atmel sendiri, tetapi di sini dimasukkan sebagai fitur security, karena ini bisa dimanfaatkan untuk meningkatkan keamanan. Adanya osilator internal di dalam mikrokontroler berarti tidak perlu penambahan kristal di luar, sehingga mengeliminasi kemungkinan penggantian kristal untuk mengubah timing dalam kerja mikrokontroler. Hal ini sebetulnya bukan hal yang istimewa, karena semua mikrokontroler AVR buatan Atmel memiliki fitur ini. Karena AT90SC merupakan mikrokontroler yang kompatibel dengan AVR maka wajar apabila AT90SC juga memiliki fitur ini. Tetapi kemampuan osilator internal di AT90SC lebih ditingkatkan lagi, yaitu bisa mencapai 20 MHz untuk CPU clock dan 40 MHz untuk cryptographic accelerator-nya, sementara di AVR maksimum clock internal hanya sampai 8 MHz.

III. Secure Microcontroller sebagai solusi Embedded System Security

Pada dua bab sebelumnya telah dibahas mengenai aspek-aspek security yang perlu diperhatikan dalam embedded system security dan tentang secure microcontroller sebagai salah satu solusi untuk embedded system security. Bab terakhir ini berisi pembahasan mengenai masalah-masalah security apa saja yang mampu diselesaikan dengan penggunaan secure microcontroller, dan masalah apa saja yang belum dapat diselesaikan oleh secure microcontroller, atau secure microcontroller saja tidak cukup untuk sepenuhnya menjadi solusi.

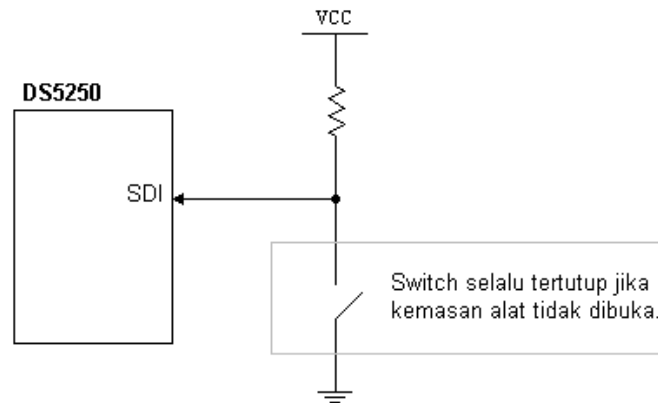
3.1. Masalah security yang dapat diselesaikan oleh Secure Microcontroller

a. Tampering

Dengan adanya berbagai sensor pada secure microcontroller, masalah tampering dapat diatasi, karena begitu terdeteksi adanya tampering, seluruh data akan hilang.

Pada DS5250 proteksi terhadap tampering lebih ditingkatkan lagi dengan adanya Self-Destruct Input (SDI). Dengan memanfaatkan SDI, proteksi terhadap tampering tidak terbatas hanya pada secure microcontroller, tetapi juga pada

rangkaian secara keseluruhan. Untuk proteksi ini, pin SDI dapat dihubungkan dengan sensor luar yang akan mendeteksi terjadinya tampering. Atau, secara sangat sederhana hal ini dapat dilakukan dengan mudah menggunakan mekanisme switch biasa. Misalnya, dibuat rangkaian switch sederhana berikut:



Switch pada gambar rangkaian di atas bisa diimplementasikan misalnya sebagai dua logam konduktor pada dua bagian kemasan alat. Dalam keadaan normal dua konduktor tersebut selalu menempel, sehingga pin SDI tidak aktif karena terhubung dengan GND (pin SDI active-high). Ketika tampering dilakukan, kedua konduktor tersebut terpisah dan tegangan pada pin SDI akan menjadi sebesar V_{CC} , akibatnya SDI menjadi aktif dan self-destruct terjadi.

AT90SC tidak menawarkan proteksi dengan SDI seperti di atas, namun demikian hal tersebut bukannya tidak mungkin untuk dilakukan. Untuk menggantikan pin SDI tersebut, bisa saja digunakan pin I/O biasa, tetapi software harus membaca pin tersebut secara berkala. Solusi yang lebih baik tentu saja dengan menggunakan pin *external interrupt*. Begitu tampering terjadi, interrupt akan terjadi dan dapat dilakukan proses “pengrusakan” yang perlu secara software.

b. Pencurian firmware

Dengan memanfaatkan enkripsi memori yang disediakan secure microcontroller, pencurian firmware dapat diatasi. Sekalipun memori program berhasil dibaca oleh penyerang, instruksi-instruksi yang tersimpan di dalamnya

tidak dapat diketahui dengan mudah karena terenkripsi. Selain instruksinya terenkripsi, alamat dari instruksi-instruksi tersebut juga terenkripsi sehingga urutannya menjadi tidak jelas.

c. Analisis terhadap bus alamat dan data serta pembacaan pin mikrokontroler

Pada pembahasan masalah-masalah security embedded system, dijelaskan kemungkinan pencurian firmware atau data di memori dengan cara mengamati bus alamat dan data. Pada secure microcontroller hal ini sangat sulit atau bahkan tidak mungkin dilakukan, karena alamat dan data tersebut terenkripsi. Pada DS5250 kemungkinan serangan ini diperkecil lagi dengan adanya vector RAM dan Dummy Bus Access. Pada AT90SC hal ini memang tidak mungkin terjadi karena AT90SC tidak mendukung pengaksesan memori eksternal. Apabila pembuat embedded system membutuhkan memori yang besar, ia dapat memilih jenis AT90SC dengan ukuran memori yang sesuai (keluarga AT90SC terdiri dari berbagai jenis dengan spesifikasi yang berbeda).

Probing terhadap pin-pin di secure microcontroller untuk melakukan analisis input dan output juga tidak dimungkinkan karena seperti sudah dijelaskan DS5250 dan AT90SC dapat mendeteksi probing. Apabila probing terdeteksi maka seluruh data akan hilang.

d. Kesulitan pada proses kriptografi

Karena keterbatasan resource dan kemampuan pemrosesannya, proses enkripsi dan dekripsi pada kriptografi menjadi sulit untuk dibuat. Dengan adanya dukungan hardware untuk kriptografi yang dimiliki secure microcontroller, masalah ini dapat dihilangkan. DS5250 dan AT90SC memiliki DES engine, jadi untuk proses enkripsi dan dekripsi DES dan triple-DES, programmer dapat menyerahkan proses sepenuhnya pada hardware. Untuk kriptografi lain misalnya RSA, DS5250 dan AT90SC memberikan sarana untuk mempercepat proses enkripsi dan dekripsinya. Pada DS5250 hal ini adalah berupa engine untuk melakukan proses aritmatika modulus yang ada pada proses enkripsi dan dekripsi RSA. AT90SC juga mendukung DH, DSA, bahkan ECC.

e. Eksekusi instruksi yang tidak seharusnya

Seperti sudah dijelaskan juga, dengan cara tertentu penyerang mungkin membuat mikrokontroler mengerjakan instruksi “palsu” untuk melakukan sesuatu yang tidak diinginkan. Walaupun hal ini hampir tidak mungkin dilakukan dengan adanya enkripsi terhadap memori program, kemungkinan ini semakin diperkecil lagi dengan fitur seperti watchdog timer.

f. Pengubahan tegangan

Secure microcontroller memiliki sensor tegangan sehingga apabila terdeteksi adanya perubahan tegangan yang ekstrim bisa terjadi pengrusakan data. Dengan demikian secure microcontroller juga mampu melindungi sistem dari serangan dengan mengubah tegangan.

g. Active timing attack

Kemungkinan perubahan kecepatan mikrokontroler untuk mengacaukan kerja sistem atau memudahkan analisis dapat dihilangkan pada AT90SC. Hal ini karena AT90SC memiliki osilator internal sehingga tidak perlu penambahan kristal di luar mikrokontroler. Jadi penyerang tidak dapat mengganti kristal untuk mengubah kecepatan mikrokontroler.

3.2. Masalah security yang tidak dapat diselesaikan hanya dengan Secure Microcontroller

a. Cost dan constraint waktu

Secure microcontroller mampu menjawab masalah keterbatasan resource dan kemampuan proses untuk mengimplementasikan kriptografi, tetapi masih belum mengatasi kebutuhan tersebut untuk implementasi security lainnya, misalnya proses otentikasi, otorisasi, dan sebagainya. Selain itu, penggunaan secure microcontroller sendiri bisa dianggap sebagai penambahan biaya, karena tentu saja harga secure microcontroller lebih mahal dibandingkan mikrokontroler biasa yang sejenis.

Solusi terhadap masalah ini bisa dikatakan tidak akan ada yang benar-benar memuaskan, tetapi pembuat embedded system harus selalu memperhitungkan tambahan biaya yang diperlukan untuk security dengan resiko yang harus dibayar apabila masalah security terjadi.

Selain itu secure microcontroller juga belum bisa menjawab masalah apabila suatu real-time system terganggu kerjanya. Jika misalnya seorang penyerang memberikan serangan DoS pada real-time system, kemungkinan besar kerja sistem tetap akan terganggu dan akibatnya bisa fatal. Penanganan lebih lanjut masih diperlukan untuk menangani masalah-masalah ini, baik secara software maupun hardware.

b. Pencurian skematik

Walaupun secure microcontroller mampu mendeteksi tampering dan apabila terdeteksi maka data akan dihapus, penyerang masih dapat melihat rangkaian dan menggambar skematisnya. Hal ini bisa dilakukan untuk melakukan analisis terhadap rangkaian dan menentukan titik yang akan diserang pada jenis embedded system yang sama.

Kelihatannya tidak ada solusi yang benar-benar mampu mengatasi masalah ini. Namun demikian resiko dari serangan ini dapat diperkecil dengan misalnya lebih banyak melakukan proses terutama bagian yang kritical pada program di mikrokontroler, bukan dari rangkaian luar. Tetapi tentu saja banyak juga bagian rangkaian yang tidak bisa dialihkan ke program. Pada kenyataannya kebanyakan pembuat embedded system memang selalu melakukan hal ini dengan tujuan untuk mengurangi cost dari sistem yang dihasilkan.

c. Serangan pada rangkaian di luar mikrokontroler

Secure microcontroller mampu mengamankan dirinya baik dari serangan secara digital maupun dari analisis misalnya probing. Namun demikian hal ini masih bisa dilakukan terhadap rangkaian di luar mikrokontroler. Penyerang masih mungkin melakukan modifikasi terhadap rangkaian untuk menyebabkan hal-hal yang tidak diinginkan. Kemungkinan serangan dengan modifikasi ini dapat

diminimalkan dengan mekanisme proteksi terhadap tampering yang baik, sehingga ketika penyerang dapat mengakses rangkaian untuk dimodifikasi sistem sudah tidak berjalan dengan semestinya dan semua data penting sudah tidak ada.

Seperti yang sudah dijelaskan, penyerang juga bisa melakukan I/O Scan terhadap PLD atau FPGA untuk mendapatkan karakteristik input dan outputnya. Serangan I/O Scan ini mungkin bisa dihindari dengan menggunakan pin I/O yang tidak digunakan pada PLD atau FPGA sebagai semacam detektor, tetapi hal ini mungkin tidak mudah juga.

d. Serangan melalui *external interface*

Kemungkinan dilakukannya serangan melalui external interface juga masih ada walaupun menggunakan secure microcontroller. Namun demikian masalah *eavesdropping* pada komunikasi dengan devais lain dapat diperkecil dengan adanya cryptographic engine pada secure microcontroller. Hal ini dilakukan dengan mengenkripsi data yang akan dikirimkan, mendeskripsi kembali data yang diterima, seperti yang memang sudah sering dilakukan pada komunikasi yang sifatnya rahasia. Namun demikian hal ini tidak selalu dapat dilakukan. Teknik ini bisa dilakukan jika embedded system berkomunikasi dengan embedded system sejenis, atau dengan sistem komputer yang memang protokol komunikasinya diamankan dengan kriptografi. Hal ini akan sangat berbeda jika misalnya embedded system tersebut berhubungan dengan devais yang lebih umum, misalnya telepon selular yang protokolnya telah diketahui umum, dan dari ponselnya sendiri tidak mendukung enkripsi.

Serangan terhadap sistem dengan menggunakan external interface yang ada tanpa melibatkan komunikasi juga masih mungkin terjadi, misalnya serangan melalui socket JTAG seperti yang sudah disinggung pada pembahasan sebelumnya. Pada kasus seperti ini mungkin tidak ada solusi yang benar-benar memuaskan (kecuali proteksi terhadap tampering yang sangat baik). Semua interface yang tidak digunakan pada saat operasinya, seperti socket JTAG yang digunakan untuk debugging atau socket untuk *In-System Programming* untuk memprogram mikrokontroler harus dibuang pada produk akhir.

e. Battery attack, passive timing attack, dan Inferensi Elektromagnetik

Secure microcontroller dapat mengatasi serangan terhadap power supply dengan cara mengubah tegangan, tetapi belum dapat menjawab masalah *battery attack*, yaitu serangan untuk menghabiskan energi yang ada seperti yang sudah diilustrasikan sebelumnya. Dengan terus-menerus membuat mikrokontroler bekerja dan mengkonsumsi daya yang tinggi, baterai dapat dibuat agar habis sebelum waktu yang diperkirakan. Kelihatannya untuk masalah ini juga tidak ada solusi yang benar-benar memuaskan, kecuali menggunakan sumber tegangan lain yang tidak rentan terhadap *battery attack*, misalnya sumber tegangan AC 220 volt yang dikonversi menjadi tegangan DC yang diperlukan dengan menggunakan rangkaian power supply. Tentu saja solusi ini sangat tidak berguna untuk banyak embedded system yang harus beroperasi pada keadaan tanpa sumber tegangan tersebut. Pada kenyataannya tanpa ada masalah security pun perancang embedded system akan lebih memilih sumber daya tersebut dibandingkan baterai apabila memungkinkan.

AT90SC mampu mengatasi masalah active timing attack, tetapi kerja sistem masih dapat dianalisis dengan melihat pewaktuan secara pasif yaitu dengan mengukur waktu. Mengatasi masalah ini mungkin akan sulit, kecuali membuat kerja sistem lebih kompleks dan sulit diprediksi dengan melihat waktunya. Tetapi hal ini bisa jadi tidak diinginkan karena tentu saja lebih diinginkan proses yang sesederhana mungkin pada sistem yang dibuat.

Secure microcontroller juga tidak mengatasi kemungkinan serangan dengan inferensi elektromagnetik (EMI). Seandainya secure microcontroller yang digunakan dibuat agar tahan terhadap EMI luar dan tidak mengeluarkan emisi EMI sekalipun, serangan ini masih mungkin dilakukan apabila komponen-komponen lain dalam sistem masih memungkinkan masalah ini terjadi. Walaupun mungkin menghilangkan EMI sama sekali sangat sulit, ada banyak cara untuk mengurangi EMI.

Pengurangan EMI bisa dilakukan pada desain rangkaian atau pada desain PCB (*Printed Circuit Board*). Pada rangkaian misalnya dilakukan dengan penambahan kapasitor. Pada PCB bisa dilakukan dengan misalnya menambahkan

copper pour (lapisan tembaga yang berfungsi sebagai *shield*). Ada beberapa cara yang sifatnya elektronis untuk mengurangi EMI.

f. Firmware upgrade

Masalah firmware upgrade seperti yang sudah dibahas juga masih mungkin terjadi pada secure microcontroller, walaupun secure microcontroller memberikan pengamanan ekstra terhadap firmware sehingga virus mungkin akan lebih kesulitan untuk melakukan perubahan pada firmware. Tetapi karena kemungkinan untuk mengubah firmware itu ada maka kemungkinan firmware itu diubah oleh pihak yang salah tetap ada. Selain itu reverse engineering terhadap firmware yang baru juga tetap mungkin dilakukan.

Sama seperti banyak masalah lainnya, masalah ini juga tidak memiliki solusi yang 100% efektif, kecuali memproteksi perubahan firmware sepenuhnya sehingga firmware upgrade tidak dimungkinkan. Namun demikian solusi seperti ini mungkin tidak diinginkan.

g. Masalah-masalah yang juga ada pada sistem komputer lain

Secure microcontroller juga tidak dapat mengatasi masalah-masalah lain seperti yang berhubungan dengan pemrograman, jaringan, social engineering, dan sebagainya. Bagaimanapun juga hal ini harus diatasi oleh programmer, pembuat sistem, dan pengguna alatnya sendiri. Dapat dikatakan hal ini memang tidak menjadi tanggung jawab dari secure microcontroller.

IV. Kesimpulan

Dengan semakin luasnya penggunaan embedded system, aspek security dalam embedded system menjadi semakin penting. Karena embedded system tidak sama dengan sistem komputer lainnya maka masalah-masalah security yang dihadapi juga berbeda, dan terdapat pertimbangan-pertimbangan yang harus diperhatikan juga.

Masalah-masalah security pada embedded system meliputi masalah keterbatasan resource dan kemampuan pemrosesan data, tampering dengan akses

langsung secara fisik, modifikasi rangkaian dan analisis secara digital, serangan secara elektrik, serta masalah-masalah lain yang tidak spesifik pada embedded system.

Sebagian masalah tersebut dapat diatasi dengan menggunakan secure microcontroller, seperti DS5250 buatan Dallas Semiconductor dan AT90SC buatan Atmel. Karena program dan data dari secure microcontroller dilindungi dengan enkripsi, analisis terhadap program dan data menjadi tidak dimungkinkan. Dengan adanya sensor-sensor pada secure microcontroller, masalah tampering juga dapat dihindari karena tampering akan mengakibatkan kerusakan data. Berbagai masalah serangan elektrik seperti perubahan tegangan dan kecepatan clock juga dapat diatasi dengan penggunaan secure microcontroller.

Namun demikian secure microcontroller bukanlah solusi tunggal untuk masalah security pada embedded system. Masalah-masalah lain seperti biaya, analisis rangkaian, serangan secara elektrik lain seperti EMI dan battery attack, firmware upgrade, dan masalah-masalah security lainnya masih mungkin terjadi pada sistem yang menggunakan secure microcontroller.

Karena itu dalam mendesain sebuah embedded system, aspek security harus menjadi pertimbangan. Segala kemungkinan serangan yang dapat dilakukan harus dapat diatasi. Secure microcontroller dapat dijadikan pertimbangan sebagai solusi yang cukup efektif untuk mengamankan embedded system, tetapi perlu diperhatikan juga bahwa secure microcontroller saja tidak cukup untuk sepenuhnya mengamankan sistem. Walaupun demikian tetap perlu diingat bahwa biaya untuk penambahan security tetap harus diperhitungkan, dan bagaimanapun juga sebuah sistem tidak akan bisa 100% aman dari serangan. Yang perlu dilakukan adalah membuat keamanan tersebut sedemikian sehingga *effort* (usaha) yang diperlukan untuk membobol keamanan sistem lebih besar dari keuntungan yang didapatkan dari keberhasilan serangan yang dilakukan.

V. Referensi

- [1] Atmel, *AT90SC12836RCFT Datasheet Summary*, 28 Juni 2005, http://www.atmel.com/dyn/resources/prod_documents/6515s.pdf
- [2] Atmel, “Secure Your Embedded Device”, *Atmel Secure Microcontroller Application Note*, 17 Mei 2006, http://www.atmel.com/dyn/resources/prod_documents/doc6528.pdf
- [3] Dallas Semiconductor, *DS5250 Datasheet*, 2003, <http://pdfserv.maxim-ic.com/en/ds/DS5250-DS5250F.pdf>
- [4] Dallas Semiconductor, “Increasing System Security by Using the DS5250 as a Secure Coprocessor”, *DS5250 Application Note 3294*, 30 Juli 2004, <http://www.maxim-ic.com.cn/pdfserv/en/an/AN3294.pdf>
- [5] Dallas Semiconductor, *Secure Microcontroller User Guide*, http://www.systronix.com/Dallas/secure_userguide.pdf
- [6] Dallas Semiconductor, “Security in Embedded System”, *DS5250 Application Note 3824*, 9 Mei 2006, http://www.maxim-ic.com/appnotes.cfm/appnote_number/3824
- [7] Dany Nativel, Atmel, “Limiting Illegal Hardware Copies by Using Secure Hardware Authentication”, http://www.atmel.com/journal/documents/issue4/pg24_25_LimitedUn.pdf
- [8] Joe Grand, “Practical Secure Hardware Design for Embedded Systems”, *Proceedings of the 2004 Embedded Systems Conference*, 23 Juni 2004, http://www.grandideastudio.com/files/security/hardware/practical_secure_hardware_design.pdf
- [9] Khalid Lateef, Ph. D., “Challenges in Embedded System’s Security”, <http://multimedia.ece.uic.edu/FIT03/Day-2/S23/p1.pdf>
- [10] Philip Koopman, Carnegie Mellon University, “Embedded System Security”, Juli 2004, http://www.ece.cmu.edu/~koopman/security/koopman04_embedded_security.pdf