Title: On evaluation and improvement of a memory integrity protection design

Introduction:

1. Security becomes increasing critical in embedded systems due to the ubiquitous e-services and consumer devices that hold and process personal and sensitive information
2. Embedded systems usually contain insecure memory components. The confidentiality and integrity of the data play an important role in the system security.
3. Using tag to identify the data is a common approach to protect data integrity. There have been noticeable efforts on design approaches for low cost tag generation and management for embedded systems. However, little systematic work has been done on evaluation of the design security.
   1. (What is the specific security issue on embedded system compared with other platform?
   2. Any work to solve these specific issues? What is their solution and what are their security evaluation?)
   3. Why we say that their work is not systematic?
4. In this paper, we target the security of a tag design which was initially proposed for the maximum resource-effectiveness. We develop a general thread model for this design and based which we identify the weakest design points that the attacker can exploit for a high attack success rate; we then propose a design enhancement to improve the design security.
   1. What is the common procedure to a systematically security analysis on a tag design?
   2. What is the expected analysis results? How to compare the results of different designs?
   3. What evaluation model do we use? What is our analysis procedure? What results do we get?
   4. Why the design enhancement can solve the security issue?

Related work outline:

1. How to do systematically evaluation of a integrity protection system
   1. The architecture of integrity protection system
      1. An hardware design based on a theoretical authentication system
   2. Threat model
      1. Common attacks on data integrity
      2. The hardware design can defend against attacks in threat model if the theoretical scheme adopted is secure
      3. Hardware design for special application
         1. Network and Cloud computing
         2. Memory data protection
            1. Uniprocessor
            2. Multi-processor
         3. Crypto-hardware design
   3. Theoretical security analysis
      1. The cryptographic primitives for theoretical integrity protection
         1. Hash
         2. MAC
         3. AREA
      2. MAC scheme
         1. Security Notion
         2. MAC schemes
            1. Deterministic MAC

Serial MAC

CBC and its variants

Parallel MAC

PMAC and its variants

* + - * 1. State MAC

GMAC and GCM

CETD

* + 1. Evaluation Methods
       1. Provable security and computational probability
       2. Formal Methods
       3. New Methods

1. The systematically evaluation of CETD

Related work

1. Data protection Typical threads to the systems, how they are modelled
   1. Threat Model:
      1. Active attack affects the integrity of data:
         1. Change the original content on memory
         2. effect the behaviour of system
      2. Introduce new content to memory:
         1. Change memory content
         2. insert new content
      3. Using existing content on the memory
         1. Copy old and replace new
      4. Q: what is the frequency of tag refreshing? A: the input of nonce is distinct, then the data-tag pair should be distinct
   2. Authentication Primitives and typical schemes
      1. Cryptographic hash function
         1. Security notions
         2. HMAC
      2. MAC
         1. Security notions
         2. Deterministic MAC
            1. Sequential MAC

CBC MAC and its variants

* + - * 1. Parallel MAC

PMAC and its variants

* + - 1. No-deter MAC
         1. GMAC
         2. CETD
    1. AERA
       1. Security notions:
       2. BL-AREA and PE-ICE
  1. Depending on the level of the systems referred. For example
     1. Cloud system and Network system
     2. Multi-processor system
        1. Using MAC
           1. SENSS

SMP protection

Based on CBC for E/A

* + - * 1. Efficient data protection for distributed shared memory multiprocessors

DMP protection

* + - * 1. I2SEMS

SMP protection

Based on GC/M

Improved version:

* + - * 1. Single-level integrity and confidentiality protection for distributed shared memory multiprocessors

DMP protection

Based on GC/M

* + - 1. Using hash
      2. Using AREA
    1. Single processor system
    2. Crypto-hardware module

1. Typical evaluation approaches
   1. Provide a theoretical evaluation to the authentication primitive used
   2. Explanatory reasoning: provable security
   3. Quantitative calculation: Computational security
   4. Formal method
   5. empirical evaluation

Our work

Conclusions

References

Motivation posted in Introduction:

1. For the tag design for embedded system, the main focus is on low cost. The related security analysis is not systematically.
2. Our previous work is low cost; the systematically security analysis is lack.

Problems to solve:

1. The concept of tag and tag design.
   1. Why tag can protect integrity
2. The security for tag design.
   1. The definition of security of tag design
   2. The approaches of security analysis
      1. Analysis procedure
      2. The results and compare
3. Some theoretical model of tag design.
   1. The motivation of design
   2. The security analysis results.
4. The application of these models on embedded system.
   1. Apply directly
   2. The original design is modified
      1. How do they modify
      2. The effect of modification
      3. Will the original security analysis hold?
      4. If not, does systematically analysis operated?
5. Original tag design
   1. Motivation of the design
   2. Do the author do security analysis? The results?
   3. The security flaw of a deisgn
6. If the motivation of this paper is analysing the security of CETD and found the inputs that enhance the collision probability of tags under replay attack, following these analysis strategy
   1. What authentication primitive it adopts
      1. Mac
      2. Nonce based, parallel
      3. Original designed MAC scheme
   2. Is this primitive analysed in theory?
      1. Self-defined security concept and results
      2. No previous methods adopted, can not compare the result with other schemes
   3. How to analyse this primitive in theory
      1. Adopt the concept of “provable security”,
      2. Use manual methods, compute the probability of tags collision under replay attack
      3. Found the inputs that lead to high tag collision probability
   4. If this primitive is secure in theory, will hardware implementation bring new security weakness?