
Hailun Tan
School of Computer Science and Engineering
The University of New South Wales, Australia
thailun@cse.unsw.edu.au

Wen Hu
ICT Centre, CSIRO
Brisbane, Australia
Wen.Hu@csiro.au

Sanjay Jha
School of Computer Science and Engineering
The University of New South Wales, Australia
sanjay@cse.unsw.edu.au

ABSTRACT
Given the limited resources and computational power of current embedded sensor devices memory protection is difficult to achieve and generally unavailable. Hence, the buffer overflow that is used by the worm attacks in the Internet can be easily exploited to inject malicious code into Wireless Sensor Networks (WSNs). We designed a hardware-based remote attestation protocol to counter the buffer overflow attack. In our attestation protocol, each sensor node is equipped with a Trusted Platform Module (TPM) board. The TPM is responsible for content verification of the program flash. To the best of our knowledge, it is the first remote attestation protocol in WSNs with each sensor node equipped with TPM.

Categories and Subject Descriptors
C.2.2 [Computer-Communication Networks]: Network Protocols; D.4.6 [Operating Systems]: Security and Protection

General Terms
Security

Keywords
remote attestation, Trusted Platform Module, Wireless Sensor Networks

1. INTRODUCTION
Although security in Wireless Sensor Networks (WSNs) has remained an active research area, a potentially more severe attack called sensor worm attack has yet to be fully studied. At first, the attacker would need to capture one sensor node physically to check if there is any software vulnerability to exploit in application. For example, due to lack of the memory protection in embedded devices, the attacker could send a malicious packet with the payload whose size is larger than the payload buffer array the application developer set aside in a processing routine upon packet reception. The extra data overwrites adjacent memory, which could include the program variable or return address of functions in the application. Hence, a malicious program could inject itself into the program memory and self-propagate to other nodes by exploiting the buffer overflow. Since the data memory and program flash are separated from each other in the Harvard architecture, adopted in all AVR micro-controllers (e.g., Atmel128 in micaZ or Atmel1281 in Fleck[1]), it is widely believed that it is not practical, if not impossible, to launch the sensor worm attack. However, recent research has shown that the attacker could still exploit the self-programming routine in the bootloader so that the malicious code could be copied from the data memory to program flash [2].

2. BACKGROUND
Trusted Platform Module (TPM) is a piece of tamper-proof hardware following specification proposed by Trusted Computing Group (TCG)[3]. Each TPM device is initialized with an Endorsement Key (EK) - an RSA public/private key pair, which is burned into its non-volatile storage. Since the private part of the key pair never leaves TPM, the attacker can not get access to the EK despite node compromise. Hence, each message signed by EK must have been originated from an authentic TPM.

In addition, TPM has several Platform Configuration Registers (PCRs), which monitor the system configuration and environments. Based on PCRs, TPM provides a special type of message encryption/decryption, which called seal/unseal. The sealed message could be associated with not only the encryption key, but also the specific PCR. Once the system has been compromised, the value of the corresponding PCR would be changed. Consequently, the sealed message could not be unsealed due to the change of PCR, even though the attacker might know the correct decryption key through node compromise.

3. DESIGN AND IMPLEMENTATION
In the following description, we assume that node $A$ issues an attestation challenge (i.e., node $A$ is challenger) and node $B$ is attested (i.e., node $B$ is attestor). The attestation protocol consists of three stages: initialization, bootloader stage and application stage. Both node $A$ and node $B$ have the same procedure at initialization and bootloader stage. The initialization phase precedes deployment and hence it is safe to assume that nodes have not been compromised. The system administrator needs to load the bootloader into sensor node and pre-configures the TPM board (e.g., load
the RSA key pairs into TPM) and establish the shared secrets between A and B ($K_{AB}$, sealed in TPM with EK and PCR$_p$) and the shared secret between the B and the base station ($K_{BS}$). At bootloader stage, the bootloader (denoted as $M_b$) is running but the application (denoted as $M_p$) is not yet loaded into the program flash. The node to be attested will be reboot with the application after the execution of bootloader. We could assume the nodes are vulnerable to physical capture but not susceptible to malware injection as bootloader can not be overwritten over-the-air. The attestation response is a digest of application code concatenated with $K_{BS}$ using an algorithm like SHA-1. The resulting digest is stored into a PCR(PCR$_p$), whose value is denoted as $V_{PCR_p}$. Similarly, the digest of $M_b$ is stored into another PCR(PCR$_a$). At the application stage, the application is running on node B once the nodes are rebooted after bootloader stage. We assume that the attacker can compromise a small number of sensor nodes, exploit a software vulnerability and launch the code injection attacks on harvard-architect architecture devices. In this poster abstract, we presented a hardware-based remote attestation protocol in WSNs with the assistance of Trusted platform Module (TPM). To our knowledge, our hardware-based attestation protocol is the first one in WSNs with each sensor node equipped with the tamper-proof hardware. Our future works includes the scalability test of out attestation protocol such as multi-hop attestation.

5. REFERENCES


