Protection Techniques of Secret Information in Non-tamper Proof Devices of Smart Home Network

Abedelaziz Mohaisen[1], YoungJae Maeng[2], Joenil Kang[2], DaeHun Nyang[2], KyungHee Lee[3], Dowon Hong[1], and JongWook Han[1]

[1] Electronics and Telecommunication Research Institute, Daejeon 305-700, Korea
{a.mohaisen,dwhong,hanjw}@etri.re.kr
[2] Information Security Research Laboratory, INHA University, Incheon, Korea
{brendig,dreamx}@seclab.inha.ac.kr, nyang@inha.ac.kr
[3] Electrical and Computer Engineering Department, Suwon University, Suwon, Korea
khlee@suwon.ac.kr

Abstract. The problem of revealing secret information in home network becomes critical when a physical capture of single device or more happen where the attacker can statically analyze the entire device's memory. While the trusted platform module that assumes a tamper proof chip in each device is not a choice, we investigate other software-based solutions. This paper introduces several mechanisms and schemes in varying scenarios for secret information protection in non-tamper proof devices of the smart home environment. The mechanisms provided herein utilize the existence of several algorithms and techniques and building blocks that do not require an extra hardware while they are computation efficient on the typical home network devices. To demonstrate the value of contributions, an extensive analysis for different scenarios including security and cost estimations are provided.

Keywords: home network, secret information, authentication, code attestation, secret sharing, security and system integration.

1 Introduction

Home Network (HN) is composed of diversity of devices which have the ability to communicate with each other internally and with other devices in other networks externally through a control point (CP). In the communication-enabled HN devices, the user's input affects devices behavior in the HN model for ease of accessibility and to provide a living-convenient environment [1]. In such environment, secret information stored and maintained in each device is used for several security purposes including device authentication, encryption and/or decryption of the internally or externally exchanged data via the CP. Generally speaking, revealing the secret information in the HN may include critical damages and threats for the privacy of part or the whole of the network. Thus, HN designers and operators carefully need to maintain the stored secret information in HN

F.E. Sandnes et al. (Eds.): UIC 2008, LNCS 5061, pp. 548–562, 2008.

devices under several expected and unexpected attacking scenarios including the physical attack.

One of the solutions to prevent physical attack is performed using the Trusted Platform Module (TPM) [2]. The usage of the TPM requires integrated hardware and operating system's supports which both complicate the overall design and structure of the HN device. Technically, once the TPM is used on an HN device the device price will be greatly increased. In addition, even though the TPM provides no chance for any successful malicious attack to be performed, it is not always guaranteed and sometimes undesirable to provide a TPM for each single device in an HN with diversity of devices of ranging prices and capabilities. Based on that, we assume the TPM is not an option in HN environment.

In an HN with non-TPM devices, an attacker does not have any difficulty to extract secret information from these devices. Even though secret information is encrypted and the key for decryption is stored outside of the device, once we the key stored externally needed to be restored for a decryption operation, the device itself need to be authenticated by the external key storing device. To overcome this reiterated problem, as the TPM scenario is out of question, we introduce a solution based on secure cryptographic functions which are already in use.

In this paper, we introduce a scheme to protect secret information in a software fashion. We revise several existing building blocks (such like RSA, secret sharing, code attestation) for protecting the secret information without any additional hardware components. Our schemes utilize these blocks in cooperation with other novel schemes as well and show a satisfactory performance and resources consumption as will be shown.

On the side of the paper structure, section 2 introduces the assumptions, terms elucidations and contribution followed by our scheme in section 3 and section 4. A comparison between our introduces schemes demonstrating their advantages and limits followed by a hybrid scheme that merges both of the schemes is shown in section 5. Security evaluation and cost estimation are discussed in section 6 Finally, we conclude by concluding remarks in section 7.

2 Definitions, Assumptions and Contributions

2.1 Definitions and Building Blocks

Definition 1 (Secret Information - SI). *is a manipulated data that provides benefit to someone where revealing this data may cause damage to its owner. Also, the indirect influence of the secret information exists where it is difficult to identify the scale of potential benefit or damage. As an example, a party's communication leads other party's benefit or damage [3] therefore the exchanged data between parties is secret information at most.*

SI in this paper means information which has a direct impact on some entity when exposed. This SI can be any of the following: a private key which is used for public key infrastructure, secret key of symmetric cryptography, bio-informatics

key, or authentication key for a HN's device. Certainly, valuable information for only its existence such like private information is also SI.

Definition 2 (Software-Based protection [4]). *is a method used for twisting the the secret information by modifying or manipulating this information through statically changed software.*

In untrusted environment, an attacker can physical capture a device with full control of its resources including the memory and thus reveal any SI. Based on that, we cannot protect SI statically stored in local memory under software-based method [4]. For the same reason, we extend software-based method in which not only the locally stored information on the same device but also other networks components may affect the overall security of the locally stored SI based on the network status. For example, a device may delete the SI to minimize the possible damage if it is informed of a possible attack.

Definition 3 (RSA Cryptosystem [5]). *In RSA, let p and q be two large enough prime numbers, $n = pq$ and $\phi(n) = (p-1)(q-1)$, SK_c is chosen such that $1 < \mathsf{PK}_c < \phi(n)$ and SK_c is chosen to satisfy $\mathsf{SK}_c\mathsf{PK}_c \equiv 1 \mod (\phi(n))$ where SK_c is CP's private key and PK_c is CP's public key.*

Definition 4 (Secret Sharing [6]). *Shamir's secret sharing is defined as follows: Let $f(x)$ be a polynomial of degree $t-1$ defined as $f(x) = s_0 + a_1x^1 + a_2x^2 + \cdots + a_{t-1}x^{t-1}$ where s_0 is the initial secret. Partial secrets are computed as $f(s_1), \ldots, f(s_n)$ and distributed for different parties. The recovery of the initial secret s_0 (and therefore the polynomial $f(x)$) requires a collusion between a group of n nodes where $n \geq t$. That is, $f(x)$ can be reconstructed by the linear regression iff (if and only if) the number of linearly independent components from the construction of $f(x)$ known to a single user is as big as t or greater.*

Definition 5 (Diffie-Hellman key exchange [7]). *Let $g \in \mathbb{Z}_p$ be a generator where p is a large enough prime. A shared key between two parties P_1 and P_2 is established according to the following interaction:*

P_1		P_2
g, p		g, p
$a \leftarrow \mathbb{Z}_p$		$b \leftarrow \mathbb{Z}_p$
$A = g^a \mod p$		$B = g^b \mod p$
	$\xrightarrow{A}$	
	$\xleftarrow{B}$	
$K_A = B^a \mod p$		$K_B = A^b \mod p$

Note that $K_A = K_B = (A)^b = (g^a)^b = (B)^a = (g^b)^a = g^{ab} \mod p$.

Definition 6 (Code Attestation). *A code is verified in code attestation by comparing the code (or representative code of the code) to other copy on a trusted party. That is performed through the CP by sending an attestation command that enforce the execution of attestation code on the remote device. Examples of these attestation methods are detailed in [8,9,10].*

2.2 Assumption

Throughout this paper, the following assumptions are used. Note that these assumptions are based on the architecture of home network in Fig. 1 representing the part of unit household.

1) The devices in the home network can communicate between each other and pass operation commands to devices within the same network or other devices that are in another network. The communication with devices in other networks is carried out through the control point. The control point (CP) is a special device (or combination of devices) which is connected community network via a 10/100 Ethernet infrastructure (see Fig. 1) [1] [11,1].
2) The different devices within the same HN can communicate between each other in a single hop communication method without getting through the CP. Thus, modifying messages between two communicating parties is hard to be carried out unless one of the two parties is compromised. This assumption is rationalized according to several other networks when communicating in a single hop (e.g., wireless sensor networks [12]).
3) An adversary needs at least a period of time longer than a threshold (say, τ seconds) to compromise a device successfully [13]. Also, to analyze the contents of physically compromised HN device, the adversary need to perform this out of the scope of the HN itself (offline analysis).
4) In most cases, the CP is trustworthy but it does not hold any secret information since we cannot guarantee the CP's safety against unknown attacks. This makes the information secure even when the CP is compromised by an attacker[2].
5) The communication traffic between two devices is encrypted using session key which is obtained by Diffie-Hellman Key Exchange (def. 5) except for few specific messages as will be shown later.
6) The CP and other devices have certificates based on RSA public key cryptosystem (def. 3).

2.3 Contributions

We investigate SI protection schemes for HN considering devices without TPM. Our contribution includes four parts which are detailed as follows

- First, we introduce a novel scheme based in which CP's broadcast periodical messages to identify the different devices in the network as safe devices or not (based on whether the network is under attack or not).
- Second, we introduce a scheme based on Shamir's secret sharing [6] and the code attestation method [9] combining their security levels for distributing SI into shares among different HN devices and authenticating each share in the different devices by authenticating the holding device itself using the code attestation method [9].

[1] The CP in our case is a server connected to a gateway.
[2] Indeed, the compromise of the CP will affect the communication links to other devices in other networks passed through the CP.

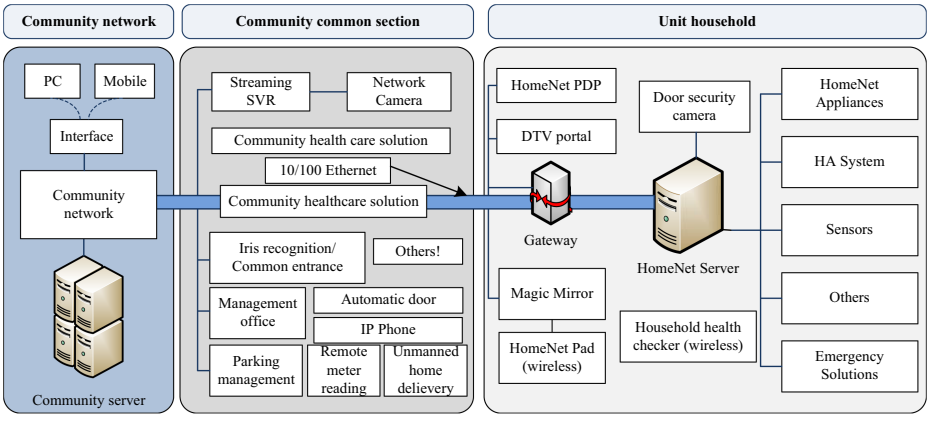

Fig. 1. The typical architecture of home network representing the premium HomeNet solution from LG Electrnonics^{TM}. The control point (CP) represents the combination of the HomeNet server and the gateway. Our proposed schemes are concerned with the unit household part.

- Third, as several advantages in each scheme comes out over the other scheme, we introduce a comparison between the two proposed schemes considering their different characteristics, advantages and disadvantages.
- Fourth and last, we merge both of the schemes mentioned above to provide a hybrid scheme to enhance the overall security of SI protection and introduce a scheme that uses both schemes merit to overcome their limits.

3 Safe Umbrella: Time-Based Scheme

The safe umbrella is our first method to protect the SI. When using the safe umbrella for protecting SI, we store the SI in volatile memory (VM) in each device (or a group of devices) in the network. Keeping SI in the memory is determined depending on the status of the device and the network with respect to attacking-probability. That means, a device erases the SI stored in its own volatile memory when it detects a possible critical attack (e.g. if the device is informed that a number of compromised devices is greater than a threshold value in a trustworthy manner). However, it is not easy to discover attacker's malicious behavior. That is, even though other devices are under attack some other devices might be unaware of this possible attack.

3.1 Overview

In the safe umbrella, we assume that each device can be informed about the network status by receiving regular message. To perform this, two scenarios are introduced. In the first scenario, a message that reports the existence of a possible attacker is generated by an HN device that informs each device to delete SI which

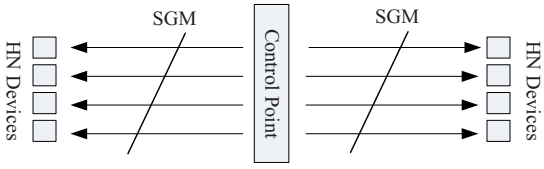

Fig. 2. Illustration of safe umbrella's

is stored in the corresponding device's memory. In the second scenario, the CP periodically broadcasts messages to inform the devices in the network about the status of the network that determines whether or not there are attackers in the network. Obviously, the first scenario is cheatable since a device under control may deviate in sending the proper signal for other devices. Other than that, there is a high probability that a device under attack cannot send the warning message on time. Thus, we use the second scenario of the solution for estimating and examining the conditions of the network.

In this scenario, we assume that the CP periodically broadcasts Safe Guarantee Message (SGM) determining the status of the network as shown in Fig. 2. Upon receiving the SGM, the different devices in the network check safety of the network indicated by checking the SGM following one of these scenarios:

- Each device deletes the SI from its volatile memory when the received SGM is not correct.
- Each device may delete the SI according to exception error generated from the execution of codes (except the case of errors related to hardware configurations). In this case, the different devices should be provided the ability to distinguish between the exception errors and relate them either to the attack possibility (malicious code running) or to normal running error (hardware exceptions).

To determine the status of the network, a time constrained verification process is held in what we call the "signed time-stamp" phase which is shown in the following section.

3.2 Signed Time-Stamp

Each device in the network can confirm its security status by the time synchronization for signed time-stamp broadcast reply. This procedure is illustrated as follows:

1. CP broadcasts signed time-stamp t_{center} signed by CP's private key to devices that CP is controlling within an estimated attack time τ where this τ period helps each device to decide whether it received the message or not. This part of procedure is performed as follows:

$$\mathsf{CP} \overset{\mathsf{broadcast}}{\longrightarrow}: D_{\mathsf{SK}_c}(t_{center})$$

where $D_{\mathsf{SK}_c}(t_{center})$ is the signed (t_{center}) using the CP's private key and the decryption algorithm D.

2. Each device checks the received time stamp t_{center} – which is decrypted by CP's public key – by comparing the decrypted one with device's own time-stamp (t_{device}) as follows (before all, the message is ignored if the decrypted t_{center} is not a valid type of time stamp):

   1: $t_{center} = E_{\mathsf{PK}_c}(D_{\mathsf{SK}_c}(t_{center}))$
   2: $\varepsilon = |t_{center} - t_{device}|$
   3: **if** $\varepsilon \leq \varepsilon_t$, network_status=safe.
   4: **else**, network_status=unsafe.
   erase SI.

3. If SGM is not received within the estimated time (i.e., ε_t), each device erases SI as a high probability of attack exists and the SGM generator is compromised.

Step 2 in the above procedure is clarified as follows: each device decides whether it is in safe status or not by checking whether the time difference between the device's local time stamp (t_{device}) and the CP's time stamp (t_{center}) is less than ε_t or not. ε_t considers the transmission delay and an additional time drift between the CP and each device. If the time difference is greater than ε_t, which indicates that an attacker probably intercepted the time packet or the CP is compromised, the device erases SI from its memory to avoid its possible reveal to the attacker.

4 Secret Sharing with Code Attestation

The above scheme, however, is vulnerable to the two possible deviation factors. First, though the CP is compromised, it may not be able to send the SGM on time. The other deviation is related to the node itself in that some possible time drift might not be considered in the design may lead to the deletion of SI though the network is not under attack.

To overcome the deviation exposed in the above time-based scheme, we introduce the secret sharing with authentication via the code attestation as solution. The scheme is detailed in the following sections.

4.1 Motivation: Necessity of Secret Sharing

Assuming that the HN devices are in an untrusted environment, we further believe that each attacker has the ability to extract the whole memory contents of any device including the SI. In the previous scheme, we store SI only in volatile memory which requires much care to maintain. However, SI is generally not for "one-time use" so that recovering the SI from a recovery remote storage system will be required. We overcome this dilemma by diffusing (i.e., distributing) SI into several devices using secret sharing to reduce the impact of single or several devices compromise. In the following, we introduce the secret sharing scheme based on the early introduced HN scenario and architecture.

Secret Sharing. In secret sharing scheme (def. 4), the SI is divided into partial secrets and the resulting shares are distributed through different devices [6]. More precisely, the SI is divided into $n \geq t$ number of partial secrets by assigning $f(i)$ to the i-th device where i is evaluated in $f(x) = s + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$ where $f(x)$ is a polynomial with random coefficients and of degree $t-1$. Devices can recover and then use the SI by recovering the coefficients of the polynomial after gathering t number of partial secrets (out of the n).

Threshold Cryptography. The main goal of the threshold cryptography is the protection of information by fault-tolerant distributing of this information among devices in the network [14]. A device can use the SI when the received shares are more than a threshold value t out of n where n is the number of the devices in the HN. Since recovering the whole secrets is possible for an attacker once he compromises a number of nodes, recovering secrets may lead to some security problems. Therefore, we use the threshold cryptography which makes usage of SI from its shares recovering it. Even though, by adapting secret sharing or threshold cryptography at the HN as shown in this procedure, we cannot eliminate all of the uprising problems. For example, since we use those methods to distribute the SI at a point, we cannot verify the device which holds the share used for recovering the partial or the whole secret. In the following, we review a direction for method that may be beneficial for the devices authentication.

4.2 Code Attestation for Shares' Authentication

Generally, authentication between two parties is performed using information that each [15] or at least one [16] of the two parties knows. However, this kind of authentication cannot be trusted in an untrusted environment because the adversary can perform authentication illegally after obtaining SI from the device which is captured physically. Therefore, received value (keys or whatsoever) for authentication

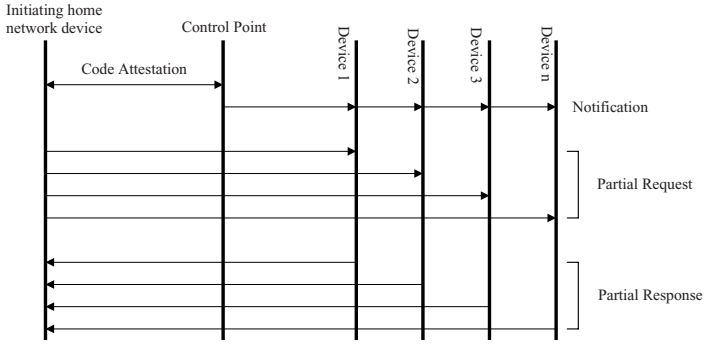

Fig. 3. Illustration of secret information recovery process in which the initiating device is authenticated via the CP which confirms the authenticity to other devices in the network. Based on the authenticity of the device, other devices may send their partial secrets to it.

might be delivered by an attacker and need to be verified (i.e., authenticated) to confirm that the value is generated by a uncompromised device.

Code Attestation: To solve the above problem, the code attestation is used (def. 6). In the code attestation, we verify a device's whole memory to detect malicious code that may exist. Therefore, this scheme requires an authenticator to have a copy of the same code of the device to emulate and to generate same value required for authentication. If the result of code attestation discovers an abnormality of modified code, the device is considered as unavailable. Code attestation is performed through the CP by sending attestation code to enforce its execution or with code that already stored in the device. It is important to exclude any probability that an attacker may inject malicious code to avoid the attestation itself. SWATT [9], PIONEER [8] and Remote Code Attestation [10] are well-known code attestation methods and can be used as a building blocks for any attestation-based scheme including our system.

4.3 Integration: Secret Sharing for Home Network

Once a device joins in to the network, the device obtains public key certificate from CP after it is verified by the CP through code attestation. The device generates public and private key and then sends the private key to CP. To reduce the computation at the side of the device, CP may generate the keys and send them to the device. The device deletes the private key after sending the key. Upon that, the following is performed:

1. CP which has SI of device generates a polynomial $f(x)$ of degree $t-1$ with t random coefficients (i.e., $a_0 \dots a_{t-1}$).
2. CP erases all of the information related to SI from the memory after distributing $f(i)$ to each devices. This guarantees that the CP is not a bottleneck anymore after the initiation phase.
3. When required, HN device requests other devices which holds partial secret to send them to recover the SI (as in Fig. 3).
4. Before the devices send the partial secrets to the initiating device, the CP verify the authenticity of the that devices and inform devices holding the partial secrets by the authentication result.
5. If the initiating device is authenticated by the CP, the CP confirms to other devices by sending response messages. The response messages are encrypted by CP's private key so that other devices in the network confirm the message by CP's public key within the validation time τ.
6. Other devices can verify the authentication response and determine whether to reply to the initiating device's request or not. The responses for the initiated requests are encrypted by session key (see def. 5).
7. The initiating device recovers the secret information from its shares (see def. 4) after decrypting the shares using the shared session key (def. 5).

Otherwise, a device requests service to CP where CP can perform the service by assembling results from the devices as threshold cryptography as in Fig. 4. CP

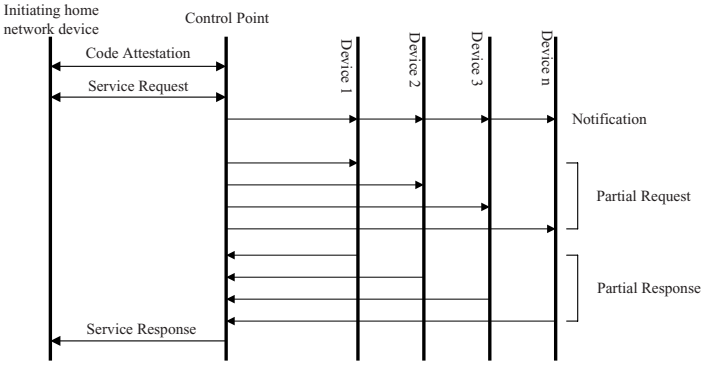

Fig. 4. Illustration of SI Merging. Unlike the recovery process, all secret merging operations from its partial shares are performed at the CP.

requests partial service requests to other devices instead of the initiating device and sends back the service response to that device. This solution, however, suffers from the trustworthiness of the CP. That is, the CP is the bottleneck for such design.

5 Comparison and Extension

5.1 Comparison

Both of Safe umbrella and secret sharing have their own characteristics which determine their advantages and disadvantages. These characteristics range from hardware implementations (i.e., memory type) to the functionality (authentication, rejoining, secret recovery, etc). Table 1 shows a brief comparison.

One of the interesting advantages of the safe umbrella over the secret sharing is the ease of the secret key usage while the secret sharing requires gathering partial secrets that require multiple message request and response as shown earlier. However, the security of the safe umbrella is heuristic in that it is determined by the attacker's ability related to the time slot available for the authentication.

Table 1. Comparison of Safe Umbrella and Secret Sharing. VM refers to volatile memory and NVM refers to the non-volatile memory.

Feature	Safe Umbrella	Secret Sharing
Device Authentication	public key of device	code of device
Storage of SI	VM of device	NVM of other devices
Rejoining Network	impossible	possible
Restoring Secret key	impossible	possible
Usage of Secret key	direct	indirect

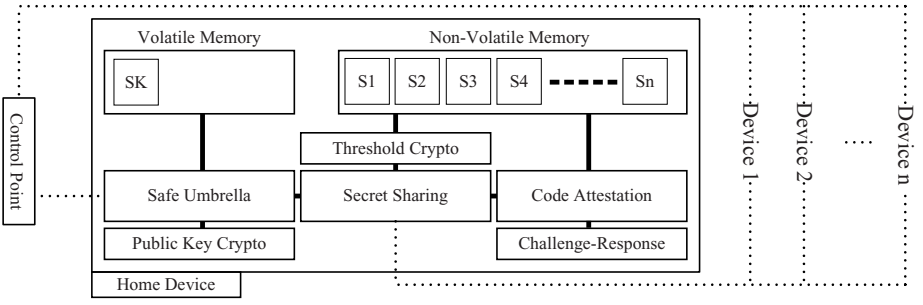

Fig. 5. Architectural view and building blocks of the hybrid scheme that makes use of both schemes' advantages

The obvious disadvantage of the same scheme, however, is that it is impossible to restore the secret key (for further use) or to rejoin to the network. On the other hand, secret sharing is provided with the rejoining feature via the code attestation and to restore the SI which makes a better solution for long-living network. However, those two features are provided on the account of using the CP which could be a bottleneck in the design.

As shown above, the safe umbrella and secret sharing has several advantageous characteristics over each other. To make use advantages of both, we briefly introduce a hybrid scheme that uses both schemes as building blocks.

5.2 Hybrid Scheme

Based on different characteristics of two introduced schemes, we provide a hybrid scheme by maintaining SI in the device's volatile memory as safe umbrella does and by enabling rejoining the network using secret sharing with code attestation. Thus, the overall procedure is as following:

1. When a device attempts to enter the network, CP performs code attestation to the device
2. If the device is authenticated successfully, the device requests from CP to restore his own secret key for the future use.
3. The device save restored secret key in its own volatile memory for future usage.
4. The CP broadcasts SGM for devices to recognize themselves as safe.
5. In case SGM is delayed over estimated time, device erases the secret key from the volatile memory.

By doing so, the different device will be re-keyed always upon joining the network. Later when the device gets valid SGM, it can again restore the secret key after attesting its memory contents. An illustration of the architecture and the building blocks of the hybrid scheme's structure is shown in Fig. 5.

6 Security Evaluation and Cost Estimation

6.1 Security Evaluation

In this section we overview and estimate the security from different points of views including the restoring chances of the SI, attack on running devices, control point's criticality, and the number of devices and that impact of that on the security.

Restoring SI. Our scheme considers the deletion of the SI stored in volatile memory when the device senses an abnormal condition (regardless to whether the revealed information is the designated SI or not)[3]. SI can be restored through secret sharing which is available only for a device which is authenticated by CP through code attestation. Code attestation is not perfect against powerful attacker who has powerful hardware, but we believe it is hard to fetch this kind of powerful attacking machine at home environment and to deceive CP for communication (with the attacker's own CP and pretending as a legal device). Finally, malicious code in a compromised device that behave as a legal device for obtaining SI will be detected by code attestation when the device tries to join in the network.

Attack on Running Device. To obtain SI on running device, an adversary must make the device sure that networking and safe umbrella are working correctly until obtaining the secret. Otherwise, adversary needs to obtain the right of execution to incapacitate safe umbrella which is impossible to be performed in the HN considering the hardness of the physical attack.

Control Point: Even though the attack on CP is difficult to be performed, an adversary can take out the information when he can control the CP. Because CP does not have SI anymore after the initiation phase, to obtain SI an attacker must disguise itself as a legal device for making a request to collect partial secrets which is hard to be done in practice is shown earlier.

A number of Devices. An adversary can restore the SI by attacking as many devices as possible in the HN. The purpose of secret sharing is dispersing attacker's target. Based on that, this attacking way is related to the cost of attack. In other words, the more devices are used for secret sharing the more security is guaranteed.

6.2 Cost Estimation

In this section we consider the overhead in terms of the computation only determined through the different building blocks of our schemes.

[3] We assume that the device can delete the SI before its physical capture due to any abnormal condition. Furthermore, we believe that reveal of the SI, in whole or in parts, to an attacker will affect security of the whole system in a way or another.

Validating SGM: RSA [5] requires higher resources compared to the symmetric key cryptography, however by choosing a small number as in many other applications (e.g., RFID and sensor networks), its operation can be faster than the normal case. Verifying message procedure is same like encrypting plain text which is decrypted by private key using public key. Therefore, having limited ability of operation, device can verify the message signed by CP's private key with a relatively small cost. To sum up, the required overhead per operation is 1 sign (at the **CP** and n verifications at the devices' side which means that a single verification is required per device.

Secret Sharing and Threshold Cryptography: Threshold cryptography requires many exponent operations. Fortunately in the proposed scheme, threshold cryptography is performed through CP with request from a device so that only the device's costs are represented in the cost of the request and response for the service. For secret sharing used to decentralize the secret key, devices require operations such like Lagrange interpolation (for polynomial). As the polynomial degree is related to the security parameter and the size of the network, the overall cost therefore increases in proportion to the number of devices.

Session Key Generation: The session key generation is required for secure exchange of the shares. The computation of the DH based key requires two exponentiation operations as shown (in def. 5).

Code Attestation. A number of operations such like hash and checksum are needed to perform code attestation. Code attestation is performed only when a device try to join the network. Even though cost for code attestation is bigger than our expectation, code attestation is rarely performed. Thus, we believe that the cost for code attestation is marginally acceptable for most of the common HN devices [4].

Since some of HN devices operate only when they are needed or scheduled by the network operator or owner, the CP may not realize whether the device is just powered off or is under attack. For this reason, CP may performs code attestation to the device for authentication when the device attempts to join in the network. The code attestation which detects malicious code on the device as an essential functionality can also be used for our authentication goal.

7 Concluding Remarks

We introduced several schemes for protecting SI in HN devices. Our schemes are based on several existing technologies and algorithms adapted into the home network environment with its specifications. Motivated by the special communication pattern in the home network, we introduced the safe umbrella which, to some extent, solves the problem of the SI protection. To overcome limits of

[4] In fact, code attestation is very applicable to sensor nodes which are essential components in the home network environment.

the safe umbrella, we introduced the secret sharing scheme with code attestation in which the secrets are maintained for long living network. To show the value of our proposed schemes, we showed detailed security and cost analysis in addition to the discussion of several network scenarios. In a future work, we will investigate other method to substitute SGM-time stamp and finding other applications for the hybrid scheme. Also, it will be beneficial to consider other attacking scenarios rather than the physical capture.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments. This research was supported by the MKE(Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement) (IITA-2008-C1090-0801-0028)

References

1. LG electronics: HomeNet (2007), `http://www.lge.com/products/homenetwork/homenetwork.jsp`
2. Group, T.C.: Trusted computing platform alliance main specification version 1.1b (2003)
3. Sengodan, S., Edlund, R.Z.L.: On securing home networks. In: INET (2001)
4. Wallach, D.S., Balfanz, D., Dean, D., Felten, E.W.: Extensible security architectures for Java. In: 16th Symposium on Operating Systems Principles, pp. 116–128 (1997)
5. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21, 120–126 (1978)
6. Shamir, A.: How to share a secret. Commun. ACM 22, 612–613 (1979)
7. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory 22, 644 (1976)
8. Seshadri, A., Luk, M., Shi, E., Perrig, A., van Doorn, L., Khosla, P.K.: Pioneer: verifying code integrity and enforcing untampered code execution on legacy systems. In: Herbert, A., Birman, K.P. (eds.) SOSP, pp. 1–16. ACM (2005)
9. Seshadri, A., Perrig, A., van Doorn, L., Khosla, P.K.: Swatt: Software-based attestation for embedded devices. In: IEEE Symposium on Security and Privacy, p. 272. IEEE Computer Society (2004)
10. Shaneck, M., Mahadevan, K., Kher, V., Kim, Y.: Remote software-based attestation for wireless sensors. In: Molva, R., Tsudik, G., Westhoff, D. (eds.) ESAS 2005. LNCS, vol. 3813, pp. 27–41. Springer, Heidelberg (2005)
11. Nakamura, M., Tanaka, A., Igaki, H., Tamada, H.: Adapting legacy home appliances to home network systems usingweb services. In: ICWS, pp. 849–858 (2006)
12. Perrig, A., Stankovic, J.A., Wagner, D.: Security in wireless sensor networks. Commun. ACM 47, 53–57 (2004)

13. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: Spins: Security protocols for sensor networks. Wireless Networks 8, 521–534 (2002)
14. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 307–315. Springer, Heidelberg (1990)
15. Bellovin, S.M., Merritt, M.: Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In: ACM Conference on Computer and Communications Security, pp. 244–250 (1993)
16. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. J. Cryptology 1, 77–94 (1988)