Secure Plug and Play Architecture for Field Devices

Nobuo Okabe, Shoichi Sakane, Atsushi Inoue and Hiroshi Esaki

***Abstract*— A PA (Process Automation) system is a kind of control systems which have been used in various non-IP (Internet Protocol) network areas. The system is now introducing IP with advantages, and will face the issues of security and configuration complexity due to upcoming new requirements. The authors have proposed a model called *Secure Plug and Play* which intends to solve these issues while satisfying restrictions, i.e. small embedded devices, isolated networks, private naming system/name space and application transparency, which are required when introducing new functionality into the existing systems. This paper shows the practicability of the model through implementing the model experimentally.**

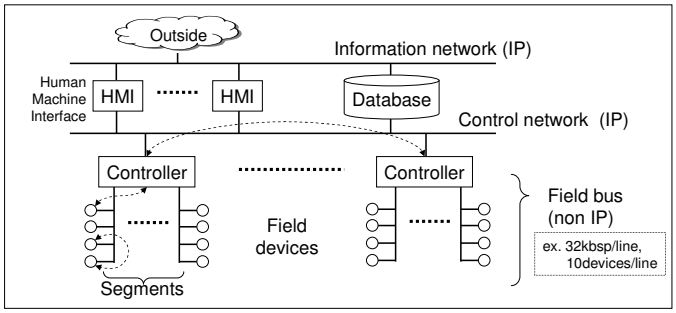

Fig. 1. A current PA system

I. Introduction

A. Current Systems

Current PA (Process Automation) systems have three regions, i.e. information, control and field (see Fig. 1). Huge systems, such as [1] and [2], are composed of several hundred controllers and tens of thousands of field devices. The field region uses neither Ethernet nor IP (Internet Protocol) as network technology and its bandwidth is limited, e.g. 32K bps. Consequently, the number of field devices per segment is small, e.g. about a dozen, and there are numerous segments, e.g. several thousand cables. The system of [1] controls a single huge petrochemical plant ($3.4km^2$), and the system of [2] [1] controls geographically distributed natural gas plants ($863km^2$).

The systems must be operated in a well-managed manner, and devices must not be connected in an ad-hoc manner.

There are multiple standards for PA systems [3], such as FOUNDATION Fieldbus (FF) [2], PROFIBUS [3] and Vnet/IP [4] They defines the systems comprehensively, e.g. functionality, layer structure, data, API and protocols, and use unicast and multicast for device-to-device, device-to-controller and controller-to-controller communications.

B. Ethernet and Internet Protocol Technology Deployment

Ethernet and IP became competitive network technology due to their huge markets. In PA systems, they are deployed to the control region, but not yet to the field region. It will bring the following advantages to the systems if they can be deployed to the field region. Please note that Ethernet can not be deployed to a part of field region due to explosion-proof environment. However, IP can be deployed to there potentially. It is described in Section V.

Enhancing functionality and performance: First, the systems can use existing and proven technology of IP, e.g. security, and/or Ethernet, e.g. redundancy. Second, Ethernet's bandwidth and performance enable intelligent field devices, e.g. handling video/audio data and uploading more precise diagnostic data. Third, IP can provide potential extendibility for the systems. The systems will be able to use the latest link technology easily because it is difficult for most of the link technologies to ignore IP's huge market. It has no impact against the application layer to change link technology because IP provides the application layer with the link layer transparency.

Reducing wiring cost: The bandwidth and capability of Ethernet can help to reduce the wiring cost of the systems (see Fig. 2). First, a single Ethernet can aggregate a large number of field buses because the bandwidth of Ethernet is greater than the bandwidth of the current field bus, e.g. 1Gbps vs. 32kbps. It gives controllers freedom from the restriction of port density. Second, the control region and a part of the field region can also be integrated on the same link physically whereas the both links are separated logically. It can reduce the cost of network infrastructure.

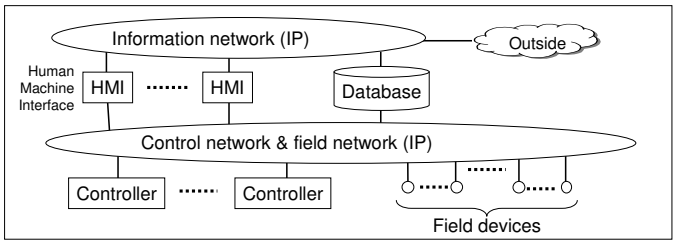

Fig. 2. Ethernet and IP integrating the control/field regions

Being more flexible: With IP, the integrated region can be extended from a single link to arbitrary networks. It gives the

N. Okabe and S. Sakane are with Yokogawa Electric Corporation.
A. Inoue is with Toshiba Corporation.
H. Esaki is with The University of Tokyo.

[1]The following document shows the system structure: http://www.mikrocentrum.nl/FilesPage/3462/Presentatie%20C3-1.pdf. STORK-GLT appeared in the document is the consortium for operating the system of NAM.

[2]FOUNDATION fieldbus is a registered trademark of the Fieldbus Foundation.

[3]PROFIBUS is a registered trademark of PROFIBUS International.

[4]Vnet/IP is a registered trademark of Yokogawa Electric Corporation.

systems the following benefits. First, it is easy to integrate and distribute the systems horizontally. Second, upper systems, e.g. enterprise systems, can access PA systems easily because both systems use the same network technology, and do not have to rely on a protocol converter. Consequently, it can contribute vertical integration.

The current field region uses FF H1 and PROFIBUS which are extended with IP and/or Ethernet. FF HSE transports application payload of FF H1 with Ethernet and IP as it focuses on PA. PROFINET uses Ethernet, but introduces its own network layer protocol rather than IP for real-time use because its target range of the response time is broad, i.e. from PA (the hundred micro-second order), FA (ten micro-second order) to motion control (micro-second order). This paper assumes that the field region uses IP as the network layer, e.g. FF HSE and Vnet/IP, since both Ethernet and IP will be important players in future PA systems, as described above, and IP can meet the real-time requirement of the systems.

This paper shows issues which PA systems will face in the future in Section II, a proposal for the issues in Section III, estimation of the proposed solution in Section VI, further study items in Section V, related work in Section VII and conclusion in Section VII.

II. Issues and Restrictions

The following are issues which the PA systems will face in the future.

Network security: The common idea of network security is to rely on the firewall model, which assumes specific network topology. However, recent incidents of computer virus show that the firewall model is not always a complete solution. It is changeable to manage security of normative devices with firewalls. Wireless technologies can expose network traffic behind firewalls easily. Therefore, end-to-end security mechanisms which do not need to assume any specific network topology are necessary.

Configuration complexity: Data, e.g. application programs and configuration information, is installed into a device in advance. Some devices will be more intelligent. It can increase the variety and complexity of the pre-installed data. And, the number of devices will increase because more precise measurement and control are required. Those will raise the cost of engineering if a plug-and-play mechanism, which enables devices to bootstrap autonomously, is not introduced into devices. This mechanism also makes recovery from a broken device easier. However, this mechanism must not be vulnerable.

The following are restrictions when introducing new functionality into PA systems.

Small embedded devices: The small embedded devices commonly used in the the systems have limited computational capability because of their restricted requirements of cost, physical size and power consumption. Some devices will have more powerful CPUs in the future. At the same time, low-power CPUs will survive because choice of CPU depends upon not only cost performance but also power consumption which has an impact against battery operation or bus width, which has an impact on circuit size.

Isolated network environment: The systems should not always require connectivity to the Internet even though introducing IP. It is the user's choice whether to connect to the Internet. Hence, functions introduced into them have to work well under an isolated network environment.

Private naming system and private name space: Information of the systems, not only the traffic but also the device's name, has to be confidential, because the information can help to indicate corporate activities, e.g. the capability of plants. Hence, the naming system should be closed to the public if operators desire. It is also important not to force a device's identity to be global unique if most of the devices should not be accessed from outside. For the above two reasons, DNS is not an appropriate naming system for the systems.

Application Transparency: From the viewpoint of the network layer, a PA system, including FF and PROFIBUS, is an application. Valuable experience and know-how have been accumulated in the application. The impact on the application should be minimal when introducing new functionality into the systems. Consequently, application transparency is important for new functionality to inherit the existing property.

III. Secure Plug and Play

The authors proposed a model called *Secure Plug & Play* [4]. This model is designed to solve the issues and satisfies the restrictions described in Section II.

A. Plug-and-play using a Directory Service

To simplify the configuration process, the model provides the device's application layer with a plug-and-play mechanism. The basic ideas of the mechanism are 1) to minimize pre-installed information in a device, and 2) to acquire most information from servers located in networks. IP address configurations are beyond the scope of this mechanism. It can be done by DHCP (Dynamic Host Configuration Protocol) in IPv4 or RFC2462 in IPv6, with which the mechanism can be combined. The mechanism requires not only name/address resolution like DNS but also general data handling, e.g. searching, getting and updating data. The authors introduce a directory service named PS (Property Server) [5]. The following are the features of PS.

- It is not a prerequisite condition for PS to connect to the Internet because PS does not require global tree structures like DNS.
- PS maintains a device's attributes as metadata of the device's identity. A typical example is that an IP address IP_{FOO} is an attribute value of an attribute type $ATTR_{IPaddress}$, and the attribute, i.e. the type and the value, belongs to a device's identity FOO as metadata.
- PS supports two types of transactions, i.e. PUT and GET. PUT sets/updates attributes in PS. GET acquires attributes from PS. Any request of transaction has search conditions which designate attributes to be affected. For example, identity/IP address resolution is done by GET transaction, where search conditions are the value of $ATTR_{IPaddress}$ belonging to the identity FOO, which returns IP address(es) IP_{FOO}.

- PS's protocol uses XML for the future extension.
- In the proposed model, every node belonging to a system has to use the security mechanism described in Section III-B. Illegal access to PS from outside can be prohibited with IPsec security policy simply. An access control list can be introduced into PS if accurate restrictions are required.

If a system has already had a directory service, it should be considered to migrate PS's functionality to the existing directory service.

B. Network Security

The authors have previously studied a security mechanism [6] which can satisfy the restrictions described in Section I, and apply this mechanism to the proposed model. The following are the features of the security mechanism (see Fig. 3).

- This mechanism provides end-to-end security. Most PA systems rely on a firewall model which assumes specific network topology. However, end-to-end security will be necessary because wireless technology and nomadic devices break the firewall model.
- Communication is protected by IPsec [7] which provides IP packets with confidentiality, integrity and authentication with the other end. IPsec is useful because its enforcement is independent from applications. IPsec is applicable to small embedded devices due to not using public key cryptography.
- It is important for IPsec to share a secret, which is called IPsec SA (Security Association), between both ends. Key exchange protocols will be important if running IPsec on small embedded devices because these devices do not have a powerful user interface like a PC, which makes manual keying difficult. For the key exchange protocol, the mechanism uses not IKE (the Internet Key Exchange) [8] but KINK (Kerberized Internet Negotiation of Keys) [9] the authors standardized. IKE is the most popular key exchange protocol for IPsec. However, it is not suited to small embedded devices because the Diffie-Hellman key exchange is mandatory. KINK can work well on small embedded devices because KINK is based upon Kerberos [5] [10], where public key cryptography is not mandated.

C. Bootstrap Sequence using the Chain of Trust

To make the plug-and-play mechanism secure, a device has to discover a trusted PS, then exchange data with PS under secure communication channels. The following bootstrap sequence called the Chain of Trust satisfies the above requirements (See A through E of Fig. 4).

1) A device can trust Kerberos server KDC (Key Distribution Center) by sharing a key. It is a prerequisite condition of Kerberos.
2) The Device should trust PS which trusted KDC shows.

[5]Kerberos is a trademark of the Massachusetts Institute of Technology (MIT).

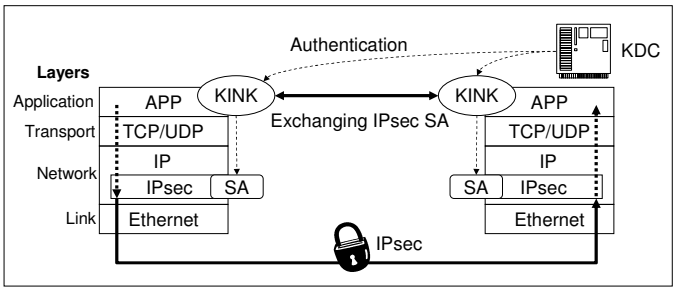

Fig. 3. IPsec security architecture

3) The device registers its information, e.g. an identity and IP address(es), to trusted PS. The information will be used for discovering peers (see Section III-D).
4) The device should trust data which trusted PS provides. Then the device can complete the sequence.

Hence, the minimum information with which a device has to be pre-installed is an identity and a key shared with KDC. Other information can be acquired from PS under a secure communication channel.

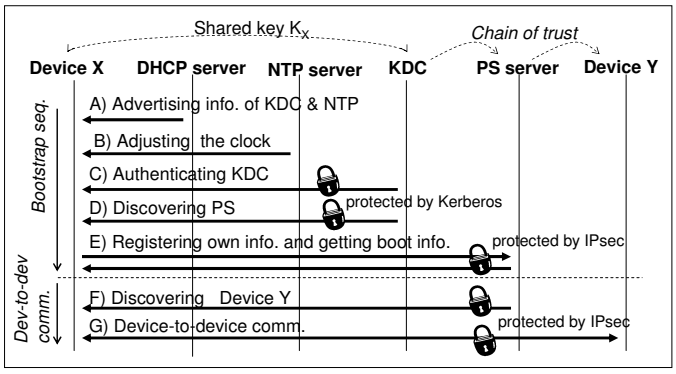

Fig. 4. The sequence of *Secure Plug and Play*

D. Device-to-Device Communication

A device and a controller have to discover trusted peers, then exchange messages with them under secure channels. The Chain of Trust can be applied in that case (See F through G of Fig. 4).

1) A device searches its peers using PS because it already knows trusted PS. (see Section III-C).
2) The device should trust peers which trusted PS provides.

PS can provide the device with the access control information as bootstrap data (see Section III-C) before starting device-to-device communication if accurate restrictions are required.

IV. Estimation

The authors implemented the proposed model to examine its practicability, i.e. object code size and performance, experimentally. Table I shows the specifications of an experimentally implemented device, whose CPU is H8/3029 (Renesas Technology Corp.), which has cryptographic hardware in a Xilinx's

FPGA. Renesas's H8 family is a popular embedded CPU in Japan. Table II shows the specifications of servers which were used for the prototyped system. The prototype system can use AES and SHA-1 instead of 3DES and MD5.

TABLE I
SPECIFICATIONS OF THE DEVICE

H/W	H8/3029@20MHz, Crypt H/W@20MHz (3DES,MD5)
OS, IP	Toppers FI4 w/ Original IP stack
IPsec	ESP (3DES-CBC,HMAC-MD5)
Kerberos	Original code based RFC4120 (etype:des-cbc-md5)
KINK	Original code based RFC4430

TABLE II
SPECIFICATIONS OF THE SERVERS

DHCP	CPU:pentium-III@1.2GHz, MEM:128MB, OS:freebsd4.10R
NTP, KDC	CPU:pentium-III@750Mhz, MEM:896MB, OS:linux2.6.8, Kerberos:Heimdal-0.6.2, KINK:racoon2
PS	CPU:celeron@1.7GHz, MEM:1GB, OS:linux2.6.8.1, KINK:racoon2

Table III shows the code size of the device.

TABLE III
OBJECT CODE SIZE OF THE DEVICE

Module	**Size** (k bytes)	**Module**	**Size** (k bytes)	**Total** (k bytes)
OS	64	Kerberos	25	256
IP (v4/v6)	132	KINK	20	
IPsec	8	Crypt	7	

Table IV shows the initial overhead on the device. The values without parentheses are net processing times, and the values in parentheses are waiting times from sending a request till receiving a reply. Those values exclude the processing time of IP address configurations, i.e. DHCP in IPv4 or RFC2462 in IPv6, and L2 address resolution, i.e. ARP (Address Resolution Protocol) in IPv4 or ND (Neighbor Discovery) in IPv6.

TABLE IV
INITIAL OVERHEAD ON THE DEVICE

	Initiator (m sec)	**Responder** (m sec)
Bootstrap	511 (282)	-
Device-to-Device	165 (125)	73

Overhead of the bootstrap sequence described in Section III-C (see A through E of Fig. 4) requires 793m sec. The total of bootstrap overheads will be several hours for a huge system described in Section I, i.e. tens of thousands of field devices. However, the entire startup time of the huge system should be longer, e.g. several days or weeks. Furthermore, the proposed model can reduce engineering tasks and help in recovering from accidents. Hence, the overhead of the sequence can be acceptable. Burst accesses to servers will be a matter for consideration if the system has a large number of devices. For the device, randomly delayed bootstrap can be a solution. However, the necessity and validity are future study items. For the server, redundancy can be a solution, i.e. the redundancies of KDCs and PSs. It is not difficult to make KDC redundant [11]. But it is a further study item for PS.

Overhead of the device-to-device communication described in Section III-D (see F through G of Fig. 4) requires 290m sec on an initiator and 73m sec on a responder. The overhead happens when both the device starts and the lifetime of Kerberos's ticket or IPsec SA is expired. The former case can be acceptable for the reasons stated in the previous paragraph. The latter case should be considered because the response time of PA systems should usually be on the hundred micro-second order. However, the overhead can be acceptable if the lifetimes are long enough, e.g. days, weeks or months, and are tuned operationally. The overhead of the responder can also be acceptable because it is shorter than the initiator's. Another possible way is to introduce priority into the IP packet processing of a device. For example, the overhead described above will have less impact if the IP stack of the device has fast-path and slow-path, and application packets are assigned to fast-path and other packets including Kerberos and KINK are assigned to slow-path. This can be a further study item.

Once IPsec is established between devices, the performance of IPsec is one of the major factors. The processing time of IPsec ESP (Encapsulating Security Payload) on the device is 4.8μ sec/byte. As an example of IPsec's throughput, the processing time for 1024-byte payload takes 5m sec. Considering the response time, i.e. the hundred micro-second order, it is fast enough.

V. RELATED WORK

There are various models of device's plug-and-play. This section shows differences between the model proposed in this paper and others.

FF HSE and PROFINET have download messages, by which a controller can set configuration data to a field device remotely. Those messages and their sequences are not protected directly because they adopted the firewall model as network security. The proposed model provides an end-to-end security mechanism and a secure sequence for the device's autonomous bootstrap which can be suited to small embedded devices and coexist with the firewall model.

[12] proposed a mechanism by which devices can acquire access control information automatically and securely. It has the following features. 1) It assumes that a device is either a client or a servers, 2) a central server provides the devices with the access control information and 3) the devices have to use public key cryptography. For the model the authors proposed, 1) the model does not have to classify a device into a client or a server, 2) PS provides the devices with any data including the access control information and 3) the security mechanism uses only symmetric key cryptography suited to the small embedded devices.

Jini [6] has the following features: a) Engaged distributed

[6]Jini is a trademark of Sun Microsystems, Inc.

object technology, e.g. RMI, CORBA, SOAP, can distribute service entities over networks. b) Servers named Lookup Service manage objects named Proxy. A client has to load an appropriate Proxy when using a distributed service remotely. c) Lookup Service, which is necessary for registering a Proxy by a service entity and for loading a Proxy by a client, can be discovered on demand with IP multicast. Jini may be suited to the purpose at which the proposed model aims since both models can provide any required data for the devices. The early version of Jini had security issues, e.g. [13], then Jini v2 [14] enhanced it. However, it is difficult to compare the actual security mechanism of Jini with that of the proposed model because it is hidden away from the specification by Java Class. Hence, Jini's applicability to devices cannot be identified. Jini v2 introduced Trust Verifier by which a client can verify the integrity of a loaded Proxy. The idea of Trust Verifier can be useful for the proposed model because the proposed model can also provide program code for the devices remotely.

In UPnP [7] (Universal Plug and Play) [15], applications have to use SOAP (Simple Object Access Protocol) and GENA (General Event Notification Architecture) which are transported by HTTP and TCP, whereas existing PA systems usually use mainly UDP. This means that existing applications will have to be changed if introducing UPnP. Hence, the goal of UPnP is different from the proposed model because one of the goals of the model is to minimize the impact on them when introducing a plug-and-play mechanism. Public key cryptography is mandated for UPnP. So UPnP's applicability to devices is also different from the proposed model.

Bonjour [8] is based on DNS-SD (DNS based Service Discovery) [16], which uses multicast DNS [17] when discovering a service on the link-local. Otherwise, DNS-SD uses conventional unicast DNS where DNS-LLQ (Long-Lived Query) [18] is proposed for more efficient alternative to polling DNS server. To simplify implementation cost on the device, the proposed model uses the same mechanism for service discovery whatever the link is. For security, DNS-SD assumes DNSSEC [19], which provides the authenticity of reply by public key cryptography. The proposed model provides mutual authentication, confidentiality and integrity of both query and reply by symmetric key cryptography. Therefore, both models have different applicability to devices. Furthermore, PS in the proposed model can apply access control to queries due to mutual authentication.

VI. Further Study Items

Kerberos's Inter-realm Operation: It is common for management to divide a large system into small manageable domains, which are called realms in the manner of Kerberos. In the case of Kerberos, inter-realm is the technique to federate operational realms by sharing a secret between KDCs (see Fig. 5). However, inter-realm has issues when being applied to PA systems as follows. 1) Inter-realm costs a device when speaking to another device belonging to a different realm because the device has to exchange messages with KDCs. 2) Host centric fashion mentioned above can be a cause of inconsistency if a system has a huge number of and a variety of devices because traversing realms is not defined precisely, but is an implementation matter. 3) Inter-realm is formed by chaining KDCs. The device will be unable to traverse KDCs even if one of the intermediate KDCs is unavailable. The authors are studying to solve these issues [20], [21].

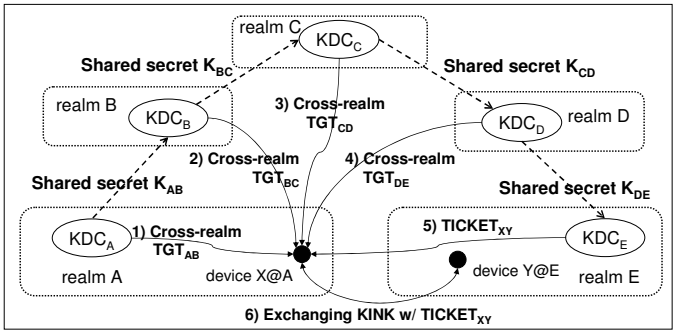

Fig. 5. An example of inter-realm

Multicast security: The following are typical multicast applications of PA systems (see Fig. 6) where each device becomes a sender and receiver, i.e. a many-to-many relationship. a) Query/reply: Multicast is used for query/reply between devices. b) Periodical advertisement: Controllers and devices advertise their information, e.g. state or heart-beat or schedule, periodically. Receivers are some or all devices in a segment. The sender does not expect any reply from receivers. c) Asynchronous command or notification: For example, a controller multicasts a command in case of emergency, and its measured data or event to multiple listeners. Multicast security is as important as unicast security whereas multicast security is more complicated than unicast security, e.g. [22], [23]. The authors are studying to extend the proposed model for supporting multicast security [24], where a device can pay the small penalty of processing overhead and memory.

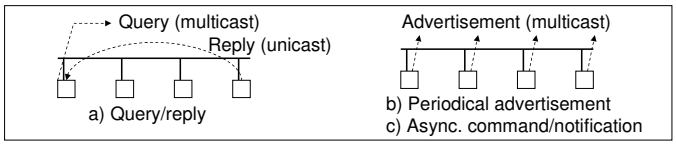

Fig. 6. Typical multicast applications

Integrating the model with existing field bus technology: A part of the proposed model's functionality may overlap with existing field bus technology, such as FF HSE. For example, they have their own device-discovery mechanisms and download messages. It is necessary to solve the overlaps for integrating the proposed model with existing field bus protocol.

IP over explosion-proof link technology: Ethernet can not be used in an entire system due to explosion-proof environment. This means that existing explosion-proof link technology, such as link technology of FFH1 and PROFIBUS, is also used even if Ethernet and IP are deployed to the field region. There are two approaches for that case. The first is to use current

[7]UPnP is a trademark of the UPnP Implementers Corporation.

[8]Bonjour is a trademark of Apple Computer, Inc.

field bus technology, e.g. FF H1 or PROFIBUS (see a in Fig. 7). It requires an application gateway named linking device by FF HSE or proxy by PROFINET. The application gateway has to be modified if a new function is introduced in the application layer. The second is to introduce explosion-proof link technology instead Ethernet under the IP layer. It requires a kind of router (see b in Fig. 7). The router is simpler than the application gateway because any change of the application layer does not affect the router. Consequently, the latter case is worth considering. One of the difficulties of the second approach is caused by restrictions of explosion-proof link technology, e.g. bandwidth, speed, frame size and power consumption. IETF 6lowpan (IPv6 over Low Power Wireless PAN) WG [25] whose purpose is to accommodate IP packets with low-power wireless personal area network, such as IEEE 802.15.4, may give a hint because both links have similar restrictions except wire/wireless.

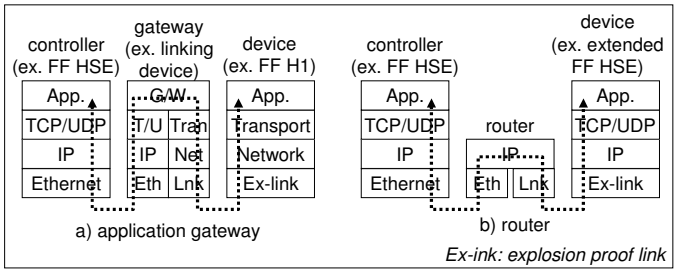

Fig. 7. Two approaches for supporting current link technology

VII. Conclusion

Through implementing the model on an embedded CPU experimentally, this paper shows the practicability of the proposed model *Secure Plug and Play* which is intended to solve the issues, i.e. security and configuration complexity, while satisfying the restrictions, i.e. small embedded devices, isolated networks, private name space/naming system and application transparency. Autonomous bootstrap sequence and device-to-device communication using the chain of trust and standardized security mechanism are the points of the model.

The device's object code size is 256k bytes which includes OS, IP protocol stack (IPv4, IPv6, TCP and UDP) and security (IPsec, the client part of Kerberos and KINK). On the device, the overhead takes 511m sec when bootstrapping, and 165m sec or 73m sec when initiating device-to-device communication. Both overheads can meet the real-time requirement of PA systems because they occur infrequently and can be tuned operationally. The device's communication is protected per-packet with IPsec whose overhead is 4.8μ sec/byte on the device. It is also reasonably small for the real-time requirement of PA systems.

There are several further study items. The first is redundancy of the servers, i.e. PS. The second is priority processing in the IP protocol stack. The third is to optimize inter-realm operation of Kerberos for small embedded devices. The fourth is to extend the proposed model for multicast security. The fifth is to integrate the proposed model with current field bus technology. Finally, the sixth is to consider existing explosion-proof link technology to be worked with IP.

Acknowledgment

The authors thank Masahiro Ishiyama, Ken'ichi Kamada and Kazunori Miyazawa for their valuable comments.

References

[1] VOGEL LIFE SCIENCE MEDIA, "All in hand at Nanhai: Control systems design at the giant Chinese petrochemicals complex," PROCESS-Worldwide, Jan. 2006, http://www.process-worldwide.com/fachartikel/pw_fachartikel_2699276.html.

[2] Nederlandse Aardolie Maatschappij (NAM), http://www.nam.nl/.

[3] *IEC 61784-1 Digital Data Communication for Measurement and Control - Part1: Profile sets for continuous and discrete manufacturing to fieldbus use in industrial control systems*, IEC, May 2003.

[4] N. Okabe, S. Sakane, K. Miyazawa, K. Kamada, A. Inoue, M. Ishiyama, and H. Esaki, "Implementing a Secure Autonomous Bootstrap Mechanism for Control Networks," *The IEICE Transaction on Information and Systems*, vol. E89-D, pp. 2822–2830, Dec. 2006.

[5] K. Kubo, J. Murakami, and T. Hoshi, "Hybrid Peer-to-Peer System for Network Monitoring of Field 'Devices," in *SICE Annual Conference 2003*, Aug. 2003, pp. 2057–2061.

[6] N. Okabe, S. Sakane, K. Miyazawa, K. Kamada, A. Inoue, and M. Ishiyama, "Security Architecture for Control Networks using IPsec and KINK," *The IEICE Transaction on Communications*, vol. J88-B, pp. 1910–1921, Nov. 2005.

[7] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC2401, 1998.

[8] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," RFC2409, Nov. 1998.

[9] S. Sakane, K. Kamada, M. Thomas, and J. Vihuber, "Kerberized Internet Negotiation of Keys (KINK)," RFC4430, March 2006.

[10] C. Neuman, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)," RFC4120, July 2005.

[11] C. Kaufman, R. Perlman, and M. Speciner, *Network Security: Private Communication in a Public World*. Prentice Hall, 1995, ch. 10.

[12] M. Naedele, "An Access Control Protocol for Embedded Devices," in *2006 IEEE Interrnational Conference on Industrial Informatics (INDIN'06)*, Aug. 2006, pp. 565–569.

[13] P. Eronen and P. Nikander, "Decentralized Jini Security," in *Network and Distributed System Security Symposium (NDSS01)*, Feb. 2001.

[14] *Jini Specifications Archive - v2.1*, Sun Microsystems, Inc., Oct. 2005.

[15] *UPnP Device Architecture 1.0, Version 1.0.1*, UPnP Forum, 2003.

[16] S. Cheshire and M. Krochmal, "DNS-Based Service Discovery," draft-cheshire-dnsext-dns-sd-04, Aug. 2006.

[17] S. Cheshire and M. Krochmal, "Multicast DNS," draft-cheshire-multicastdns-06, Aug. 2006.

[18] S. Cheshire, M. Krochmal, and K. Sekar, "DNS Long-Lived Queries," draft-sekar-dns-llq-01, Aug. 2006.

[19] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," RFC4033, Mar. 2005.

[20] N. Okabe, S. Sakane, K. Kubo, K. Miyazawa, and K. Kamada, "Optimizing Ineter-realm Operation for Control Networks," in *3rd International Conference on Network sensing System (INSS 2006)*, June 2006, pp. 68–73.

[21] S. Sakane, S. Zrelli, and M. Ishiyama, "Problem statement on the cross-realm operation of Kerberos in a specific system," draft-sakane-krb-cross-problem-statement-01, Oct. 2006.

[22] P. Kruus, "A survey of multicast security issues and architectures," in *Proc. 21st National Information Systems Security Conference*, 1998.

[23] T. Hardjono and B. Weis, "The Multicast Group Security Architecture," RFC3740, Mar. 2004.

[24] N. Okabe, S. Sakane, K. Miyazawa, and K. Kamada, "Extending a Secure Autonomous Bootstrap Mechanism to Multicast Security," in *The 2007 Symposium on Applications and the Internet (SAINT 2007) Workshop*, Jan. 2007 (to appear).

[25] IETF 6lowpan WG, http://www.ietf.org/html.charters/6lowpan-charter.html/.