Key Management for Substations: Symmetric Keys, Public Keys or No Keys?

Shailendra Fuloria[1] ,Ross Anderson[1], Fernando Alvarez[2], Kevin McGrath[2]
[1]Cambridge University, [2]ABB Inc.

***Abstract*—In this paper, we discuss symmetric-key and public-key protocols for key management in electricity transmission and distribution substations—both for communication within substations, and between substations and the network control center. Key management in the electricity network is widely regarded as a challenging problem, not only because of the scale, but also due to the fact that any mechanism must be implemented in resource-constrained environments. NISTIR 7628, the foundation document for the architecture of the US Smart Grid, mentions key management as one of the most important research areas, and the IEC 62351 standards committee has already initiated a new specification dedicated to key management.**

In this document, we describe different variants of symmetric-key and public-key protocols. Our design is motivated by the need to keep the mechanism simple, robust, usable and still cost effective. It is important to take into account the complexity and the costs involved not just in the initial bootstrapping of trust but also in subsequent key management operations like key update and revocation. It is vital to determine the complexity and the cost of recovery mechanisms—recovery not only from malicious, targeted attacks but also from unintentional failures. We present a detailed threat model, analysing a range of scenarios from physical intrusion through disloyal maintenance personnel to supply-chain attacks, network intrusions and attacks on central systems. We conclude that while using cryptography to secure wide area communication between the substation and the network control center brings noticeable benefits, the benefits of using cryptography within the substation bay are much less obvious; we expect that any such use will be essentially for compliance. The protocols presented in this paper are informed by this threat model and are optimised for robustness, including simplicity, usability and cost.

***Index Terms*—key management, substation communication**

I. Introduction

Key management for substation communication has gained salience as a research problem with NISTIR 7628 [6] and IEC 62351 [5] standard series highlighting its importance. If some utilities and other customers decide to authenticate communications between intelligent electronic devices (IEDs) in electricity substations and the station control computer / micro-SCADA server (hereinafter the 'substation controller'), we will need mechanisms to load initial key material into controllers and IEDs, update this key material from time to time, and recover from various errors and attacks. In this paper, we sketch both symmetric-key and public-key mechanisms, discuss the possible failure modes and attacks, and compare the costs and benefits of the two approaches.

We assume that the station controller will have a TLS certificate to communicate securely with the network control center (NCC) in line with IEC 62351 [3]. But as well as the controller, a substation might have several hundred IEDs. If the customer wishes to secure the internal communications for reasons of regulatory compliance, or communicate directly to some IEDs from the NCC or from another remote location without these communications going through the controller or the gateway, then the IEDs must also be furnished with key material. We need to consider two cases – where a new substation is being set up, and where one or more IEDs are being added, replaced or re-keyed in a working substation. Our objective is to minimize the total lifecycle cost of key management from design, manufacture and acceptance testing through operations.

In the rest of the document, we will discuss how a new IED is added to the substation. At time of manufacture, the same process is simply repeated for each IED, most probably at the start of either the factory acceptance test or the site acceptance test. From the usability point of view, the process is similar regardless of the underlying cryptographic technology. Each IED comes with an ignition key printed on its packaging. To set up the IED, the engineer enters this ignition key into the substation controller. An authentication protocol is run between the IED and the controller, which indicates successful enrolment of the IED. The actual authentication protocol is different in the symmetric-key case from the public-key case.

II. Symmetric Keys

The simplest symmetric-key protocol is that used for example in Homeplug AV [7]. The ignition key m is a 128-bit AES key with which the IED is loaded at time of manufacture and which is printed on its packaging. Once the new IED is physically connected to the substation network and the engineer has entered this ignition key m into the substation controller, the following protocol is run.

The first author's research is funded by ABB. The contents of this article do not necessarily express the views of ABB.

Shailendra Fuloria is a PhD candidate at the Computer Laboratory, University of Cambridge. (email: Shailendra.Fuloria@cl.cam.ac.uk)

Ross Anderson is a professor of Security Engineering at the Computer Laboratory, University of Cambridge. (email: Ross.Anderson@cl.cam.ac.uk)

Fernando Alvarez works with the Power Systems division of ABB at Baden, Switzerland. (email: Fernando.Alvarez@ch.abb.com)

Kevin McGrath is a scientist at the ABB Corporate Research Center, Norway. (email: Kevin.McGrath@no.abb.com)

In the first step, the IED *Y* sends a join request to the controller, encrypted under the ignition key m. There is also a random challenge *N* so the IED can check that the controller's response is not a replay. The substation controller decrypts this request and sends back a message containing the random challenge, the device serial number, a unique device key *KY*, and the network key *KN* currently in use – all encrypted under the ignition key. The IED confirms receipt by sending back *N* encrypted under *KN*. Writing *Y* for the IED and *C* for the controller, the above protocol, we have

$Y \rightarrow C$: $\{Y, N\}_m$
$C \rightarrow Y$: $\{N, Y, KY, KN\}_m$
$Y \rightarrow C$: $\{N\}_{KN}$

Once these messages have been exchanged, a green light comes on in the controller telling the engineer that the IED has been enrolled successfully. The controller now has an entry for *Y* in its key database containing *m* and *KY*.

The IED will now use the network key *KN* to authenticate communications with the controller as well as multicast messages (GOOSE and SMV) to other substation IEDs [4].

In some cases, the substation automation system may include devices that are outside the station's physical security boundary, even though they are inside its logical trust boundary. In such cases, it may be desirable to provision these devices with session keys with which they can communicate with the substation controller directly, and whose compromise would not affect the cryptosecurity of any other device. A decision on whether to provide such functionality should be made in view of the threat model discussed in section VII below.

A. Key Update

The period after which the substation keys must be updated should be a matter for the utility's security policy (which we'll touch on later). Keys might be updated regularly, or only in response to the compromise of an IED. The controller sends the IED a new network key, together with a random nonce encrypted under the unique device key *KY*; the IED confirms receipt by returning the nonce encrypted under the new network key.

$C \rightarrow Y$: $\{Y, N, KN'\}_{KY}$
$Y \rightarrow C$: $\{N\}_{KN'}$

Since the network key is shared between all the IEDs, the controller has to send it to every device on the LAN.

$C \rightarrow Y1$: $\{Y1, N, KN'\}_{KY1}$
$C \rightarrow Y2$: $\{Y2, N, KN'\}_{KY2}$
…
$C \rightarrow Yn$: $\{Yn, N, KN'\}_{Kyn}$

Finally, once the controller has received acknowledgement from enough IEDs, it announces the transition to the new network key.

B. Variant protocols

The rationale for the above protocol becomes clearer when we consider two ways to simplify it. First, we might omit the unique device key *KY*:

$Y \rightarrow C$: $\{Y, N\}_m$
$C \rightarrow Y$: $\{N, Y, KN\}_m$
$Y \rightarrow C$: $\{N\}_{KN}$

In this case, on key updating, the ignition key must used again:

$C \rightarrow Y$: $\{Y, N, KN'\}_m$
$Y \rightarrow C$: $\{N\}_{KN'}$

Second, we can dispense with the ignition key *m*. If we assume that IEDs will always be initialized in a secure environment – that the engineer is always trustworthy, that the substation LAN is always effectively isolated by its firewall from network-borne threats, and that no IEDs have been subverted – then the unique device key *KY* can be simply set up by a message from the controller. First *Y* announces its presence

$Y \rightarrow C$: Y

The controller asks the engineer if *Y* can join the network. He types 'yes' and the controller sends *Y* a device key *KY*, followed by the network key encrypted under it:

$C \rightarrow Y$: $KY, \{N, KN\}_{KY}$
$Y \rightarrow C$: $\{N\}_{KN}$

This corresponds to the 'simple connect' mode of HomePlug, which is similarly used where the risk of compromise at installation time is low. The advantage is that it saves the engineer the effort of typing in the ignition key *m*; it's plug-and-play. The disadvantage is that this procedure is dangerous if any compromised device is present on the network, as it will learn the values of *KY* and *KN*.

III. USING PUBLIC-KEY MECHANISMS

In this approach, the ignition key *m* printed on the device packaging is not an AES key but is the hash of an X.509 certificate, issued by either the utility or the vendor. We assume that a public-key approach would use standard protocols and mechanisms such as TLS, which is not only the standard for protecting communications between the station controller and the NCC but is also used in web browsers to protect e-commerce and online banking [1]. It has been formally verified to be secure and many implementations are available. It has several variants. While e-commerce and banking sites typically use a combination of server certificates and user passwords, it is commonly assumed that substation automation would use both client and server certificates. In that case, the TLS authentication protocol runs as follows.

1. The client sends a *client hello* message to the server, which contains its name C, a transaction serial number C#, and a random nonce N_c.

2. The server sends a *server hello* message to the client, which contains its name S, a transaction serial number S#, and a random nonce N_s.
3. The server sends a *server certificate* message with its certificate CS containing its public key KS. The client verifies this certificate against the root certificate issued by the certifying authority.
4. The server sends a *server hello done* message and a *certificate request* to the client.
5. The client sends a *client certificate* message with its certificate CC containing its public key KC, followed by a *client key exchange* message containing a pre-master secret key K_0 encrypted under the server's public key.
6. The client sends a *certificate verify* message that is signed by the client's private key and contains the *master secret* K_1 (hash of the pre-master key with the nonces sent by the client and the server). The server verifies the signature. From here onwards, all the messages are in encrypted form. We denote this as $\{\ldots\}_{KCS}$ in the client-server direction and $\{\ldots\}_{KSC}$ from the server to the client.
7. The client sends a *finished* message with a message authentication code (MAC) computed on all the messages to date. The key for this MAC is the *master-secret* K_1; the session keys *KSC* and *KCS* are derived by hashing the master secret with the nonces.
8. The server sends a *finished* message with a MAC computed on all the messages to date.

We can write this in standard protocol notation as

$C \rightarrow S$:	$C, C\#, N_c$
$S \rightarrow C$:	$S, S\#, N_s$
$S \rightarrow C$:	CS
$S \rightarrow C$:	Certificate Request
$S \rightarrow C$:	Server hello done
$C \rightarrow S$:	CC
$C \rightarrow S$:	$\{K_0\}_{CS}$
$C \rightarrow S$:	$\{K_1\}_{CC^{-1}}$
$C \rightarrow S$:	$\{\text{finished}, \text{MAC}(K_1, \text{everything\_to\_date})\}_{KCS}$
$S \rightarrow C$:	$\{\text{finished}, \text{MAC}(K_1, \text{everything\_to\_date})\}_{KSC}, \{\text{data}\}_{KSC}$

One suggestion is to use this authentication protocol straight out of the box in substation automation with the IED as the client and the substation controller as the server. The network key *KN* would then be shared as the 'data' in the last message of the above protocol.

This is a more heavyweight mechanism than the simple shared-key protocol discussed earlier, and it brings with it a number of costs and constraints. First, all IEDs must have sufficiently capable processors to do public-key cryptographic operations (an ARM is fine, an 8051 is not).

A second issue is the cost and complexity may depend on the process by which client devices acquire a certificate. There are four options here: provide each device with a utility certificate when it is purchased; provide the certificate when it's installed; outsource certificate management to a commercial certification authority such as Verisign; and do without client certificates altogether.

A. Certificate provided at time of purchase

When the utility purchases an IED from a vendor, it is brought to a key management facility at the utility or at a specialist contractor. This facility provides the device *Y* with a public key *KY*, a corresponding private key KY^{-1}, a certificate $cert_u(KY)$ signed with the utility's key, and an ignition key *m* which is the hash of the certificate

$$m = hash(cert_u(KY))$$

Installation proceeds as before; this time the station controller ends up with a copy of the IED certificate in its key database rather than *KY*.

B. Certificate provided at installation time

In this scenario, the new IED comes with a vendor certificate, plus the hash of this certificate printed on the packaging.

$$m = hash(cert_v(KY))$$

Installation proceeds as before, with the engineer entering *m* into the substation controller. The IED now sends its certificate to the controller, which forwards this to the NCC in a secure TLS session. The NCC verifies the IED's certificate and generates it a new one, signed by the utility. This is sent via the controller to the IED.

$\text{Y} \rightarrow \text{C}$:	$\text{cert}_v(\text{KY})$
$\text{C} \rightarrow \text{NCC}$:	$\{\text{cert}_v(\text{KY}), \text{Y}\}_{\text{KC,NCC}}$
$\text{NCC} \rightarrow \text{C}$:	$\{\text{cert}_u(\text{KY}), \text{Y}\}_{\text{KNCC,C}}$
$\text{C} \rightarrow \text{Y}$:	$\text{cert}_u(\text{KY})$

Once the IED has got the utility certificate, things proceed as in IIIA.

C. Certificate provision outsourced

In this scenario a commercial certificate authority (CA) such as Verisign is contracted by the utility to manage the process of issuing and managing certificates. This moves much of the complexity away from the utility; however the CA industry is concentrated, leading players extract monopoly rents, and there may be complex issues around sovereignty, liability and the protection of national infrastructure.

A further issue is the integration of certificates (or other key material) with the customer's asset register. The typical utility has a database of installed IEDs; device registration and key certification might economically be integrated, and similarly certificate revocation is likely to be engineered in line with existing control procedures (which we discuss in section VI)

D. Server Certificates only

The cost and complexity of managing certificates can be reduced (though not eliminated) by using only server certificates. In this option, TLS enables the station controller to operate like a secure website. The ignition key *m* can function just as a password. The engineer enters *m* into the controller, and starts up the IED, which initiates a secure session with the controller; this asks for the password *m*; if it's correct, the controller gives the IED the key *KN* as the 'data' (as explained in section III).

IV. Auditing Mechanisms

Whether or not a utility wants a mechanism to audit the bootstrapping process would depend on its security policy. If it does, then it would need a mechanism to verify that the IED installed in the substation was manufactured by the vendor, and not some rogue device secretly pushed into the supply chain. An audit mechanism could help mitigate such a supply chain attack (more details in section VI C). The actual detail would vary depending on whether symmetric key or public key based protocols are used. The utility would use a suitable protocol to verify remotely that the IED keys match the device's serial number.

V. Key Backup and Remote Access

Key backup can be complicated and difficult. It is important to recover from non-malicious failure of equipment, yet the presence of key material at multiple locations can make a system more vulnerable to attack.

The database at the station controller that contains the key material for the station IEDs can be backed up at the NCC. There is a further aspect in that if some customers require direct remote access to individual IEDs then copies of the IEDs' keys need to be available at some central key distribution centre or certification authority, which we assume will be co-located with the NCC. In the symmetric-key case, the database includes live device keys *KYi* and so it must be well protected against failure of confidentiality; in the public-key case the database contains public keys rather than the device private keys, but its authenticity should still be protected (lest an attacker introduce the certificate of a compromised device).

VI. Revocation

Revoking a key with the symmetric key protocol would simply involve a key update. If a network key needs to be revoked, the substation controller has to follow the key update protocol outlined in section IIA. As discussed earlier, the actual protocol will depend on the variant of the symmetric key protocol that the utility has decided to use.

Revocation in public key infrastructures has always been a complicated problem. The typical method is that the certifying authority maintains a central list of revoked certificates and all the devices regularly access this list.

In the substation environment however, things are not so simple. Many devices are safety critical requiring high levels of availability. Simply revoking a certificate, and stopping all communication to and from the IED, is not in general an adequate response to suspected compromise. Revocation will have to be engineered in sync with the current industry practice of dealing with breaches in substations – this could mean that the IED continues to operate till the utility finds a window to repair or replace it.

VII. Threat Vectors

Recovery from attacks is more complex; we have to consider a number of cases.

A. *Malicious intruder*

Utilities worry about intruders, who might range from teenage vandals, through members of a protest group, to saboteurs from a nation's intelligence service. Vandals and protesters might be satisfied with doing immediate physical damage that causes a service outage; government agents (and the most sophisticated protesters) might leave behind vulnerabilities to be exploited later. Our focus is on attacks that leave behind potential network exploits, such as firewall compromises or the extraction of crypto keys.

Every IED has a maintenance port that can be used to make a physical connection. Any intruder who gains access to the substation bay can use this port to access the IED and configure or reprogram it; a knowledgeable intruder might be able to extract key material. A crypto vendor might suggest fitting all IEDs with tamper-resistant crypto chips; but this would not prevent an intruder installing malicious software designed to cause a failure at some future time; and the defence of the substation LAN against network-borne attack is fundamentally down to the capability of the firewall and to the software in the controller which acts as the network gateway.

It is not clear that using cryptography to authenticate messages within the substation does much to mitigate the effects of a physical intrusion. If cryptography were not used, then the maintenance staff would simply get the equipment back in working order, reload all the software, test the systems and secure the perimeter once more. If cryptography is used, staff must go to the extra effort of rekeying devices. (This is one reason why in the symmetric-key case we use device keys *KCi* rather than just ignition keys *m*; the former can be replaced after a compromise.)

B. *Malicious repair staff*

An even worse case arises when a member of the maintenance staff is discovered to have acted maliciously, or comes under suspicion. Again, there's a range of seriousness, from a disgruntled employee who is fired following misconduct, to a staff member discovered to have been in the pay of a hostile intelligence service. Such a number of staff may have had physical access to a number of substations and might also have had extensive technical knowledge. The cost of recovering from such incidents may be significant, especially if the vendor is risk-averse or under a heavy regulatory or compliance burden.

Here again it is not clear that cryptography buys much in the way of protection. The first-line response to such an incident would be to identify the substations visited by the suspect and check that the security-critical software there (the firewall and the controller) had not been tampered with. Again, the use of cryptography means a small additional cost, of rekeying the networks in the affected locations. Even so, if the attacker managed to leave malicious software in an IED that was not detected in the subsequent forensic examination, that software would have access to the new keys. So the forensics are again more important than the crypto.

C. Supply Chain attacks

Both nation states and criminals may subvert supply chains; for example, PIN entry devices (PEDs) for use with point-of-sale equipment have recently been discovered with hidden mobile phones that texted customers' card and PIN details to the attacker. Given the growing interest in cyber-war, it is conceivable that a nation state will find a way to embed malicious hardware and/or software in IEDs. (Like the compromised PEDs, they are often manufactured in China for cost reasons.) How might a utility act, following the discovery of such an attack?

While properly implemented mechanisms to audit the bootstrapping process can help mitigate the risk of a rogue device ending up in a substation network, protection against more subtle attacks where the ignition key of a particular device is compromised would be more complicated. If the utility knows or suspects that the IED *Y* was compromised, the attacker will know the ignition key *m*, the private key KY^{-1} if any, and in theory all keys protected by them. In practice the situation will not be as bad, as the attacker can only decrypt those ciphertexts that he sees; if he knows the *m* that was used to protect *KY* and *KN* in $\{N, Y, KY, KN\}_m$ then so long as this ciphertext was not observed, *KY* and *KN* will still be secure. Again, it's the substation firewall that we depend on.

The case with public keys is similar. We've assumed so far that private keys in devices persist indefinitely, so the attacker can in theory get keys protected by them; but in practice he won't get the ciphertext as it remains on the substation LAN. We might redesign the protocols to replace private keys from time to time – for example under option 3.2 we might generate a new private key for the utility certificate on installation. But the design details could be tricky (we don't want the IED to choose private keys as we're assuming its software is compromised) and the benefits are uncertain (compromised software can do other bad things). Again, it's not clear that either symmetric-key or public-key crypto buys us much.

D. Attacks over the network

Most substations have a network connection to the NCC, whether dial-up, a private network or a VPN over the Internet. We are starting to see more exotic connectivity such as Bluetooth links that enable maintenance staff to access the controller from the comfort of their truck [2]. Attacks can come over any of these networks. The attacks of most concern to regulators, as to officials concerned with protecting critical national infrastructure, are those that can be performed remotely – whether by accessing substation controllers directly, or via the NCC or other enterprise systems that talk to the controllers. We have recently seen a Trojan aimed at Siemens SCADA systems. Attacks might not even be targeted; there have been incidents where flash worms infected SCADA servers and caused local service denial on the LAN because of the volumes of traffic they emitted.

The most likely attack vectors at present are zero-day exploits on network-facing machines such as firewalls, and spear-phishing attacks that use social engineering to trick staff into installing malware on critical machines. Neither of these attacks can be prevented by substation LAN cryptography, or even interact very much with it; if an IED in the substation is taken over by a Trojan once authentication is fitted, then traffic coming from the Trojan would be duly authenticated. And although an attacker who compromised a controller or NCC might siphon off the key database, there would be little direct benefit in this so long as he had access.

Again, the discovery of a network-borne attack would lead to a recovery effort whose goal would be to restore the software integrity of the system. If a machine with access to a key database had been compromised, rekeying might be considered necessary.

E. Non-malicious failure of a substation key database

Where the database in the substation that contains the cryptographic keys suffers a non-malicious failure (maybe a hardware fault), we might find that IEDs would not be able to take any further commands from the controller. As this would be a high-impact failure, we require rapid and robust recovery mechanisms. As discussed before, there are several ways to do key backup. The point here is that cryptography becomes another component that can fail; when building high-availability systems we need to provide for resilience through mechanisms for redundancy, backup, recovery and so on. The engineering costs are nontrivial, and previous experience (with cryptography supporting prepaid electricity meters in the mid-1990s) suggests that the total costs will be several times the initial estimate once proper provision has been made for resilience.

F. Compromise of the NCC key database

If the substation's IED keys were backed up at the network control center, and if IED keys are later used to allow remote access (e.g. from enterprise systems), then the NCC's key database would become a potent attack point. During the mid-1990s there was much debate on key recovery, and one of the doomsday scenarios was the compromise of a master key database such as this. The cost of recovery – if it involved rekeying tens of thousands of devices in hundreds of substations – could be substantial. Such a risk could become one of those to which the utility's insurers would pay attention, leading to expensive compliance requirements, possibly involving regular key replacement.

G. Intrusion in unprotected areas

Most substations have a robust physical boundary in the form of wire mesh or (increasingly) hard wall. However, there are smaller transformers with IEDs in unattended and unprotected locations like pole tops. Such IEDs could be a more attractive target for a technically savvy attacker.

This seems to be a more attractive scenario for the deployment of cryptography than substation LANs. The communications are in principle easy to tap, and the use of message authentication codes could stop this. But the critical problem is scalability. If attackers have to climb poles and attach wires, they can only do so many before getting caught; so they will prefer high-value targets (which are not common at the lower levels of the distribution network). Things are quite different with attacks that can be automated; if a hacker working for a protest group or a hostile state can take over tens of thousands of distribution transformers remotely, that might be attractive even if the individual targets are low-value. But again, the devil is in the detail. If the vulnerability is that devices using unauthenticated (and publicly known) protocols are protected only by the obscurity of their IP addresses, then VPNs using commercial off-the-shelf technology might be a cheaper and quicker way to fix it.

VIII. Conclusion

The cryptographic protection of control-system communications on wide-area networks is clearly a good thing, and the IEC 62351 standard provides for the use of TLS to protect the link between a substation controller and the network control centre. The development and deployment of such technology is overdue and it is to be devoutly encouraged.

The protection of communications within the substation brings much less obvious benefits. We examined the cases of symmetric-key and public-key mechanisms, and considered a range of threat scenarios from physical intrusion, through maintenance personnel subsequently found to be disloyal, through supply-chain attacks, network intrusions and attacks on central systems. Where the substation network lies entirely within protected space, an intruder can do more damage directly, and the addition of cryptographic protection merely increases the cost of recovering from incidents. Where there are outlying devices, cryptography can provide some value but may be provided better by COTS VPN products.

Given the dubious benefits of using cryptography to protect substation communications, we suspect that it will mostly be deployed for reasons of compliance rather than risk management, and suggest that key management use the lowest-cost mechanisms, namely symmetric-key cryptography as in section II or even section IIB.

IX. References

[1] R Anderson, *"Security Engineering – A Guide to building Dependable Distributed Systems"*, Second edition, Wiley 2008

[2] R Anderson, S Fuloria, *"Security Economics and Critical National Infrastructure"* presented at the 8th Workshop on Economics of Information Security (WEIS'08), at *http://www.cl.cam.ac.uk/~sf392/publications/WEIS-2009.pdf*

[3] F Cleveland, *"IEC IC57 Security Standards for the Power System's Information Infrastructure – Beyond Simple Encryption"* at *http://www.xanthus-consulting.com/Publications/*

[4] S Fuloria, R Anderson, K McGrath, K Hansen, F Alvarez, *"The Protection of Substation Communications"* in proc. of SCADA Security Scientific Symposium, Jan 2010, at *http://www.cl.cam.ac.uk/~sf392/publications/S4-2010.pdf*

[5] IEC 62351, *"Power systems management and associated information exchange – Data and communications security"*

[6] NIST, *"NISTIR 7628, Guidelines for Smart Grid Cyber Security" at http://csrc.nist.gov/publications/PubsDrafts.html*

[7] R Newman, S Gavette, L Yonge, R Anderson, *"Protecting Domestic Power-line Communications"*, Symposium On Usable Privacy and Security (SOUPS), July 2006 at *http://www.cl.cam.ac.uk/~rja14/Papers/homeplug-soupspaper.pdf*

X. Biographies

Shailendra Fuloria is a PhD candidate at the Security Group, Computer laboratory, University of Cambridge. His research interests include security of industrial control systems, security of the electricity network and critical national infrastructure. Contact him at *Shailendra.Fuloria@cl.cam.ac.uk*

Ross Anderson is the professor of security engineering at the Computer Laboratory, University of Cambridge. His research interests range from security protocols and APIs through hardware tamper-resistance and critical national infrastructure to security economics and security psychology. Anderson is a fellow of the Royal Society, the Royal Academy of Engineering, the Institute of Engineering and Technology, and the Institute of Mathematics and its Applications. He's also the author of the standard textbook *Security Engineering—A Guide to Building Dependable Distributed Systems (Wiley, 2001).* Contact him at *Ross.Anderson@cl.cam.ac.uk*

Kevin McGrath is responsible for coordinating projects in the IT security domain for the Industrial Communication research program within ABB Corporate Research, with a focus on embedded system security. In addition, he is an R&D project manager for technology development projects and contributes to the development of the technology roadmaps and project portfolio within Corporate Research. Contact him at *Kevin.McGrath@no.abb.com*

Fernando Alvarez is the chief architect for defining security architecture and development strategies for ABB's Power Systems Substations. As a Security expert, he leads the technical group related to security for Substation Automation Products. Fernando is an active member of TC57 WG15. A technologist with vast experience in software and system design and implementation, he graduated from California State University, Long Beach with Bachelors in Computer Science Engineering. Contact him at *Fernando.Alvarez@ch.abb.com*