



Smart dust, friend or foe?—Replacing identity with configuration trust

Howard Chivers *, John A. Clark

Department of Computer Science, University of York, Heslington, York YO10 5DD, UK

Available online 25 June 2004

Abstract

Smart dust motes are miniature self-contained systems that may be deployed in very large numbers. In military applications these devices are subject to different threats than conventionally deployed systems, for example, attackers may deploy counterfeit devices to subvert the integrity of a system, and this may be a greater concern than confidentiality. The possibility of identity theft reduces the value of conventional authentication methods, prompting a re-evaluation of how to achieve integrity in such systems.

This paper reviews the authentication problem from first principles, but instead of regarding each pairwise network interaction separately, it shows how to build *Configuration Trust* in a system as a whole. The new trust process focuses on establishing the type or function of a device, rather than individual identity, supported by key distribution that introduces just sufficient diversity to detect subverted motes. This is embodied in protocols that establish communication with a mote and manage the system configuration. These protocols are shown to be effective in defeating standard communication attacks, and a detailed analysis demonstrates their strength in the face of an attack by overwhelming numbers of counterfeit motes.

The paper concludes by illustrating the use of this trust mechanism in a simulated dust war. The simulation results confirm that this Configuration Trust system is robust, even under aggressive attack.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Configuration Trust; Smart dust; Authentication; Integrity; Key management

1. Introduction

Smart dust motes are miniature systems with self-contained power and communications that have the potential to provide unique new military capabilities [1], including robust sensor networks in specific locations or over wide areas of interest. They may be deployed to overcome physical

or environmental inaccessibility, for covert surveillance, or because very large numbers of sub-miniature devices are difficult to physically counter-attack.

Much attention has been given to the physical construction of these devices [2–4], resulting in the possibility of building self-contained systems in volumes under a cubic millimetre, capable of point-to-point optical communication to a base station or mobile user. Deployment strategy studies [5,6] have shown that such sensors will be effective in area surveillance.

* Corresponding author.

E-mail address: chive@cs.york.ac.uk (H. Chivers).

However, to date there has been relatively little published work on the system security aspects of such devices. If a surface of dust sensors is deployed, for example, to detect intrusion in a cleared building, or the movement of forces in a theatre of interest, then there may be a strong motivation for the surveillance target to attack the system.

The threats to such a system may be different from those to conventional networks. Operational decisions are likely to depend on sensor data so it is critically important to ensure that an attacker is not able to introduce misleading information. Integrity of sensor data is therefore more significant than its confidentiality. Denial-of-service is also a concern, but in many scenarios will draw unwelcome attention to the attacker.

This naturally raises the question of how a system user can recognise authentic dust motes from alien devices.¹ Systems that need to be identified do so by demonstrating possession of an Authentication Token, often in the form of a cryptographic key. However, the ease with which a resourceful enemy could capture motes makes the exposure of some identities inevitable, so other mechanisms are needed to provide assurance in system integrity.

System assurance [7] methods have been developed to provide confidence in security properties such as integrity; they rely on good development practice, skilled testing, and operational configuration management. However, smart dust systems are composed of large numbers of devices with uncertain location and erratic connectivity; under these circumstances it is important to reconsider how configuration control can be achieved. We use the term *Configuration Trust* to describe the whole problem of creating and managing a sub-system and identifying to the user the behaviours that can be expected. This is a wider concept than authentication, or its associated key management, since it deals with confidence in the service provided by

the network, not just the interaction between client and server.

Previous work by the current authors [8] has suggested that a critical design factor in ubiquitous systems is *fungibility*: are devices inherently so similar that they are exchangeable? This suggests that identifying dust motes by function rather than identity may be a promising approach, because it removes the need for an identity infrastructure and allows the system design to focus on the essential configuration and integrity requirements.

This approach is used here to develop a comprehensive Configuration Trust system for dust sensors. Individual dust motes are identified by type, and Configuration Trust mechanisms establish groups of similar motes whose outputs can be checked for consistency. Of course, such a system must not simply comprise a large number of replicated alien devices, so Configuration Trust is supported by a *diversity* mechanism implemented by a novel key management scheme. Each mote is provided with a Diversity Key drawn from a small set. Combined with batch deployment information this scheme is shown to be very successful in ensuring that a base station can assemble a diverse system, rendering it resilient to impersonation attacks. Because this trust mechanism can measure diversity, it also determines when particular motes are over-represented; as a result it has an inbuilt mechanism for detecting and rejecting attacks using overwhelming numbers of alien motes.

This paper introduces the concept of Configuration Trust and contributes a novel and robust scheme for its implementation in distributed sensor systems. It also demonstrates the need to critically deconstruct conventional ideas of identity and authentication in pervasive systems in order to better address system security properties.

This work is presented in four main sections. Section 3 provides a detailed account of the dust scenario and explores the requirements for Configuration Trust in such a system. Section 4 describes protocols to support these concepts and analyses their robustness in the face of standard attacks. Section 5 analyses the performance of the trust system in the face of a determined attacker who is able to capture dust, extract keys and mount a counter-attack with the aim of subverting

¹ We use the term alien device to include a whole range of communication and system attack mechanisms including subverting authentic motes, simulation of mote communication or even the manufacture and large-scale deployment of counterfeit dust motes.

the system. Finally, Section 6 confirms the effectiveness of these proposals by presenting results from a simulation of dust deployments and counter-attacks.

2. Related work

Configuration management, authentication, and identity systems to support large mobile or static facilities are well developed; public key systems are the foundation of many practical systems [9] and are also the basis of protocols used in computing grids [10]. The business community uses public key cryptography as the basis for trust services [11], providing a hierarchy of credentials that identify both servers and services. These solutions are applicable to mainstream military systems, but the physical characteristics of smart dust, and its threat environment, make their deployment impractical in this application.

The ubiquitous computer community has proposed a number of trust mechanisms. Proximity can be used for authentication, supported by time-of-flight protocols [12–14]. This paper assumes that a user is able to measure approximate location, but relative direction [3] is a more useful supporting mechanism than approximate distance in this scenario.

Several authors [15–17] have suggested that trust is a metric of the expected behaviour of a device, and proposed ways of combining trust recommendations to provide a metric for the expected outcome of an interaction. Few authors have studied the system stability of such concepts, but one study [18] has shown that convergence of trust estimates is possible, but only given a low proportion of malicious nodes.

These trust mechanisms are also slow to converge. In the commercial world this is managed by constraining transaction values during the learning period; in a military scenario it is likely to be a major weakness, since it may be necessary to place considerable reliance on the first interaction with a server. Slow convergence and lack of robustness would seem to make these trust mechanisms inappropriate for distributed sensor applications.

Researchers in ad hoc networks have proposed key management schemes, most of which elaborate the work of Eschenauer and Gligor [19], Blom [20] and Blundo et al. [21]. All these schemes deal with the problem of establishing point-to-point secure communication in an ad hoc mesh network using pre-distributed keys and symmetric cryptography. Eschenauer and Gligor distribute a random subset (a ‘key-ring’) of keys to each node, allowing point-to-point communication between nodes with common keys. Blom [20], followed by Blundo et al. [21] describe how pairwise link keys can be calculated from a set of base keys in each node. A node base is one row of a symmetric key matrix; each node has a different row, ensuring that the capture of nodes or link keys, up to a threshold, does not allow the derivation of link keys between other nodes. The most mature schemes are straightforward combinations of these two approaches [22,23], although authors have also suggested randomised pairwise keys [24] that similarly allow node identification.

These key distribution schemes are all concerned with the threat that a captured node may undermine the confidentiality of other links in the system. They fail to address the key threat in our scenario: that a subverted node may be used to undermine the integrity of data provided by the system.

One elaboration of the Eschenauer and Gligor [19] scheme uses deployment knowledge to improve key assignment [25]; deployment information could be used to further tune the system described in this paper, as noted in Section 7.

Sensors that use low power radio [2], may be forced to communicate in a mesh, rather optically to a base station. A concern of researchers in this area is the propagation of data to fusion nodes and the possibility that a data fusion node may itself act maliciously [26]. In contrast, this paper starts from the viewpoint that data fusion will take place in a base station (or ‘by a user’) and is instead concerned with sensor identification and validation; however, these results may be equally applicable in radio systems with explicit data fusion nodes.

Sensors have restricted computing power, which limits their ability to carry out cryptographic

functions. Some researchers make the case that this rules out the use of public key protocols [27], while others have shown that it is possible to reduce the on-line processing cost in the sensor, assuming a relatively powerful base station [28]. This paper does not take a position in this debate: it uses public key protocols because they are simpler and are widely understood, but since the key management described here uses pre-deployed keys, we believe that it could also be adapted to symmetric cryptography.

3. Overview of Configuration Trust

This section begins by providing an overview of the operational scenario and the associated requirements for Configuration Trust. Three key factors that can be used for authentication are then outlined (knowledge, physical attribute, behaviour), and we show how each of these can be used to provide confidence in the desired configuration. Finally, we summarise how these elements are combined, introducing the protocols that are described in Section 4.

3.1. Detailed scenario

The main focus of this paper is the deployment of smart dust in territory that is militarily hostile. Sensors may be deployed over an area, a series of tracks, or at point locations, by mechanisms as varied as hand-held dispensers, mortars or Micro-Air Vehicles (MAVs) [1]. Motes will probably be deployed in high volume to overcome unpredictable physical placements, resulting in a sensor-rich environment.

Individual motes have limited power, and the most promising method of communication is optical point-to-point, to base stations that may be local users, temporary mobile infrastructure, or even Low Earth Orbiting satellites [1]. The communication topology is therefore a set of ad hoc star connected networks, formed from in-place sensors and mobile base stations. Communication sessions will need to support both long lasting surveillance operations and the needs of mobile users who may read sensor information while in

transit. A user may be able to estimate the approximate location of a mote [3], but communication to any given device may be infeasible, or temporarily obscured.

The design of motes will include physical protection for cryptographic keys, or other private information, but the availability of devices to an attacker, in quantity, will inevitably result in the successful analysis of some devices. The system must therefore be robust in the face of technically advanced attacks that have been informed by such analysis. Important elements of the scenario are shown in Fig. 1.

The deployers' and the users' infrastructures are also significant. Both will afford more physical protection for internal keys or software than is possible for dust motes but mobile users may have limited access to a backbone communications infrastructure, either by circumstance or choice.

3.2. Configuration Trust requirements

In this scenario, desirable properties of a Configuration Trust system are

- That a user should be able to make rapid judgments on which results to trust.
- That it should not be necessary to keep track of individual sensors, either during deployment or between communications sessions.
- That the system should be robust, even assuming that an attacker has subverted or extracted private information from individual motes.

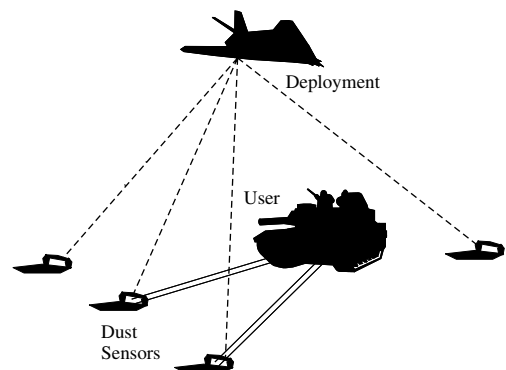


Fig. 1. Dust sensors (motes) deployed in advance in an area of interest and exploited by an independent mobile user.

- That it should be difficult for an attacker to overwhelm the system, even by deploying equivalent numbers of alien motes.

The need for a user to make a rapid judgement requires a *Stateless* configuration decision, meaning that the user may assemble a network of sensors to obtain specific information, but the decision to trust specific motes, or results, in that temporary network should not depend on information learned in previous similar interactions, either by that user, or others.

3.3. Design issues

In a conventional system, servers provide confidence in their identity by demonstrating the ownership of a cryptographic key. This arrangement is supported by an infrastructure to issue keys and ensure their validity.

Identity keys could be distributed in this scenario, but the vulnerability of individual devices make it unrealistic to assume their protection. In practice, ad hoc communications and the huge numbers of motes would also prevent the collection and distribution of validity information.

It is hard to meet the Configuration Trust requirements if individual identities can be subverted. Consider an attacker who has managed to obtain private information that could be used to prove the identity of a single mote. The attacker can simply deploy a large number of alien motes, all with the same identity. The user is likely to encounter an alien mote because of the volume of the attack, and trust it because of the stolen identity. The risk of attacks of this sort would force the user into maintaining state between interactions and prevent a user from trusting early encounters with new motes.²

The value of a compromised device to an attacker can be multiplied by manipulation of the

physical or communications environment as well as by the manufacture and deployment of alien motes. For example, spoof devices could be constructed with higher communications power or simulators could impersonate multiple devices. All these tactics enhance the likelihood of a user communicating with a subverted mote, regardless of the number of devices that the attacker has succeeded in analysing.

These considerations suggest some design objectives:

- The Configuration Trust system should force an attacker to attack the system ‘in place’. (The most expensive scenario for an attacker is for sensors to be locale-specific: if the sensor from one general location can be identified by a user as distinctive, then an attack will depend on subverting devices from the same area.)
- The physical communication pattern of dust should be distinctive. (Some forms of physical communication (see Section 3.1) allow a user to observe that the distribution of sensors is consistent with their deployment. This forces an attacker to distribute dust in order to attack dust.)

Both these objectives increase the resilience of the system: an attacker should be forced to successfully analyse many dust particles and deploy dust in overwhelming volume in the correct general location. This paper will not discuss the physical nature of communications further; the scenario assumes that some degree of location measurement is possible, so the Configuration Trust system must be able exploit whatever type location selectivity is available.

3.4. Configuration Trust factors

When people are authenticated to a system, it is good practice to base the identification upon a group of factors, rather than a single key. For example:

- Something the person knows (e.g. passphrase).
- Something physical (e.g. smart card).
- Something the person is (e.g. biometric).

² Since stateless mechanisms are the goal of this paper, we shall not detail the arms race between attacker and users that would result if the user responded with a stateful protocol. Suffice to remark that given a small number of subverted keys there are several attack and defence strategies; but the user would be forced to build trust conservatively.

Provided the factors are independent, the use of several enhances confidence in the decision. Generally, one factor serves as a primary key that can be correlated with measurable attributes of the others. In identity systems the correlation is carried out by means of a database of acceptable attributes (e.g. fingerprints). By analogy, the life-cycle of a dust mote also allows the evaluation of a number of attributes:

- Something it knows: an Authentication Token.
- Something physical: approximate location.
- Something it is: behaviour.

An *Authentication Token* serves to provide a primary key or identifier that can be linked to other authentication factors; these must have features that can be independently measured by a user. The difference between mote and human identification is that mote attributes must be evaluated by a user in a stateless protocol, rather than looked up in an identity database. Discussion of the token and its content will be deferred until the other factors have been explored.

3.5. Location

The physical communication channel will allow the approximate location of motes to be determined by the user (see Section 3.1). However, for this to be useful it must be correlated with a measurable attribute of the mote, in circumstances where the physical deployment of motes does not provide traceability.

Most high volume devices are manufactured, packed, transported and deployed in batches. The deployment mechanisms previously mentioned (hand dispenser, mortar, miniature aeroplane) will all maintain the approximate physical co-location of a batch of devices, so the inclusion of a *Batch Key*³ in a mote can provide the user with an attribute that is shared by devices in the same approximate location.

³ *Batch Key*: In the sense of a Primary key or Identifier, not necessarily a cryptographic key. Batch identification also provides valuable traceability in the event of flaws or attacks on the manufacturing or deployment chain.

A user can simply check that a device has the same Batch Key as those in the same approximate location. Assuming that the attacker is unable to manufacture Batch Keys in any way (we will justify this later) then information obtained by analysis of a mote can only be used in the general location from which the mote was obtained. Assuming that such analysis is difficult, then location binding by batch has a profound effect on the practicability and work factor of attacking a large-scale deployment.

3.6. Behaviour

Subverted identities can be deduced by observing many behaviours, either for uncharacteristic performance or to determine that an identity is over-represented. Since our objective is to provide stateless decisions (see Section 3.2), multiple observations must be made by selecting a group of motes and simultaneously comparing their performance. There are two parts to this problem, the first is how to carry out the behavioural comparison, and the second is how to ensure that the motes are different physical devices.

Comparison of sensor outputs is scenario and sensor specific, and sensor fusion is a mature subject (see, for example [6,29]) so this will not be explored here in more detail. We assume simply that a user will use a fusion function that is able to produce a correct result even if some motes in the group return corrupt values.

The other indication of identity theft, over-representation, is used as part of the configuration management protocol, and this described in detail in the sections that follow. Essentially, the fusion of the output from several motes allows the occasional subverted input to be ignored, while the detection and rejection of over-represented individuals prevents alien motes from dominating the group.

The sections that follow use the term *Comparison Group*:

A Comparison Group is a temporary, small, set of motes, whose outputs can be combined in such a way that a corrupt minority does not materially affect the desired result.

This approach relies on the assumption that the devices in a Comparison Group are different; for example, to prevent an attacker from deploying many identical alien devices. Restating the assumption: members of a Comparison Group do not need to be identified as individuals, *it is sufficient to ensure that they are all different*.

This requirement is much simpler to meet than managing mote identities; a single *Diversity Key*, drawn at random from a set of possible keys, can be assigned to each mote. The user simply ensures that each mote in a Comparison Group holds a different Diversity Key.

3.7. The Authentication Token

The previous sections have identified factors that can be used to establish trust in a group of motes: approximate location (consistent batch) and diversity. In both cases the user validates these factors by comparing the claims of individual motes in the group.

To support this process, the mote must be able to provide evidence of its Batch and Diversity Keys. The Authentication Token therefore acts as container to convey these keys to the user.

The next section describes how an Authentication Token can be constructed, but the underlying requirement is that an attacker should be unable to forge such a token.

3.8. Summary

Dust motes present opportunities for identity theft by attackers; however, it is possible to provide assurance in the behaviour of a group of motes without supporting individual identities. The group is chosen to ensure batch consistency, which correlates to consistent location, and diversity, which prevents an attacker overwhelming the group with subverted devices. Outputs from all the motes in the group are fused to ensure that a small number of subverted devices do not affect the quality of the result.

Two processes are required to support Configuration Trust:

- Lifecycle management of Authentication Tokens. (Providing a mote with an Authentication Token containing its Batch and Diversity Keys and the delivery of the token to a user.)
- Managing the membership of Comparison Groups.

The Configuration Protocol, described in the next section, shows how these processes can be implemented.

4. Protocols

The Configuration Protocol (4.2) manages the lifecycle of Authentication Tokens, and the management of Comparison Groups. This protocol is supported by a secure connection between a user and a mote. Although the Introduction Protocol that establishes this connection is a conventional public-key exchange, it will be described first for completeness.

Each protocol is followed by an informal analysis of its susceptibility to a range of standard attacks.

4.1. The Introduction Protocol

The protocol described below uses RSA [30] public key cryptography to demonstrate how communication can be set up with a sensor (see Section 2 for a discussion of the use of public key in motes).

4.1.1. Definitions

The Introduction Protocol takes place between a user (U) and a single mote (M : the smart dust sensor).

We use the following notation for public key operations: The encryption key, decryption key and modulus for principal p are, respectively: E_p , D_p , N_p . The encryption key is public, unless otherwise stated.

The encryption operation $(X)^{E_p} \bmod N_p$ is written $E_p\{X\}$.

A challenge, or nonce, R is a random number that is used as part of the protocol to ensure that

responses are fresh, and not replayed from an earlier transaction.

4.1.2. Initial state

The mote has been provided with the public key, E_u , of the user, and an Authentication Token, A_m , for the mote (see 4.2). The Authentication Token is described in more detail below, but it does not need to be unique. The mote has generated a random public key pair E_m, D_m, N_m ; its public key, E_m , has not necessarily been distributed.

The user has access to the private user decryption key, D_u .

4.1.3. Objectives

The purpose of the Introduction Protocol is to

1. Establish a secure communication link between M and U that will subsequently be used to query the sensor mote.
2. Assure M that U is an authentic user.
3. Provide M 's Authentication Token, A_m to U .

4.1.4. Protocol

The overall protocol is shown in Fig. 2. It is initiated by the user, who provides a suitable nonce:

$$U \rightarrow M^4 : R_u. \quad (4.1)$$

The mote responds by encrypting the user's nonce, the mote's public key, the mote's Authentication Token and a second nonce under the user's public key:

$$M \rightarrow U : E_u\{R_u, E_m, A_m, R_m\}. \quad (4.2)$$

The user responds with a session key and the mote's nonce encrypted under the mote's public key:

$$U \rightarrow M : E_m\{K_{\text{session}}, R_m\}. \quad (4.3)$$

The session key can then be used to protect subsequent communication. This description does not

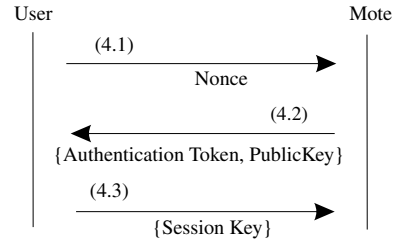


Fig. 2. Outline of the Introduction Protocol.

include service selection,⁵ but if required the second and third steps can include an offer of services and a selection.

4.1.5. Analysis of the protocol

This protocol is a straightforward public key exchange, with the property that A_m is provided only to an authentic user.

The following analysis assumes that suitable public key components and message compilation methods are used, ensuring that encrypted messages can only be constructed by a system in possession of both the content and the private key.

Replay. Step two includes a nonce from step one, and similarly step three includes a nonce from step two; it is therefore not feasible to replay the messages in the second and third steps from a previous transaction. A replay of step one results in the user replying with a valid output; this is no use to an attacker without the user's private key.

Man-in-the-middle. Without possession of the user's private key, an interceptor or active man-in-the-middle is unable to decrypt the second message, which precludes access to the mote's public key and its Authentication Token. Without the mote's private key a man-in-the-middle is unable to obtain the subsequent session key.

User spoof. The mote has been provided with the user's public key by an out-of-band trusted channel and no mechanism is provided for rekeying. A spoof user would therefore have to be in possession of the user's private key in order to operate the protocol, at which point a wider range

⁴ $U \rightarrow M$: should be read as “ U sends to M the following message”.

⁵ Service selection is used to select such features as cryptoalgorithms, session protocols, qualities of service and functional parameters.

of attacks on the system would be possible (see Section 4.2.4).

Mote spoof. We anticipate that dust will be subject to hostile analysis, and by this means a user’s public key, E_u , and an Authentication Token, A_m will become known to an attacker. This would allow an attacker to impersonate a mote by establishing a communication link and providing the user with a genuine Authentication Token. This subject is discussed in depth in Section 5. Mote impersonation would also allow the identification of authentic users. In most scenarios if an attacker can intercept such users, they can probably be distinguished, but an enemy can also distinguish friendly communicators by using a subverted authentication system.

4.2. Configuration protocol

The purpose of an Authentication Token is to provide evidence to a user about a mote. Section 3 identifies the two elements necessary to support Configuration Trust:

- A Diversity Key.
- A Batch Key.

Either key can also be used to differentiate functional or security features of the mote, by using different key sets for different types of mote.

The Diversity Key is used to ensure that different members of a Comparison Group are actually different motes. This prevents an attacker from obtaining authentication information from a single dust sensor and re-using it, either in a simulation or with multiple alien sensors. The Batch Key allows the user to judge that a sensor’s batch is consistent with its approximate location.

The system is most effective at resisting attack if an attacker is unable to separate the Batch and Diversity Keys. In this situation an attacker who has compromised a mote obtains only the combination of both types of key.

Since we assume that individual motes will be compromised, it follows that there should not be sufficient information in a mote to allow the Diversity and Batch Keys to be being extracted from the Authentication Token.

4.2.1. Objectives

The purpose of the Configuration Protocol is to

1. Ensure that Authentication Tokens are properly constructed and are provided to motes.
2. Specify how users utilise Authentication Tokens, location, and Comparison Groups to provide Configuration Trust.

4.2.2. Definitions

The keys associated with a particular mote are a Batch Key, B_i , and Diversity Key, D_j .

4.2.3. Protocol

This protocol manages the lifecycle of an Authentication Token, and is summarised in Fig. 3. At the point of manufacture, the mote is initialised with an appropriate random Batch Key:

$$\rightarrow M : B_i. \tag{4.4}$$

Before deployment the Diversity Key and User’s public key are provided to the Mote:

$$\rightarrow M : E_u, D_j. \tag{4.5}$$

The mote encrypts the Batch and Diversity Keys using the Users Public key, forming an Authentication Token. The separate Batch and Diversity keys are then deleted from the mote:

$$M : A_m = E_u\{D_j, B_i\}. \tag{4.6}$$

The Authentication Token is then passed to an authentic user as part of the Introduction Protocol (see protocol step 4.2):

$$M \rightarrow U : A_m. \tag{4.7}$$

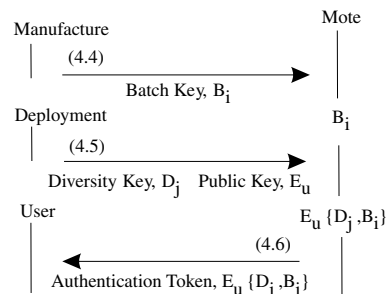


Fig. 3. The lifecycle of an Authentication Token.

The user is then able to decrypt A_m to obtain D_j and B_i .

The user is seeking to assemble a *Comparison Group*—a set of motes whose outputs can be compared or merged to provide a reliable result; the user must reject this mote as a member of the group if either:

The user has already established a link with a mote with the same Diversity Key, (4.8)

or,

The Batch Key of the mote is not consistent with the physical location of the device.

The simplest implementation is to ensure that the batch key is consistent with other group members. (4.9)

And, in addition:

If the user already has a putative member of the Comparison Group with the same Authentication Token, then both motes should be excluded from the group, and so should any further motes with the same token. (4.10)

Otherwise the mote is accepted into the Comparison Group.

4.2.4. Analysis of the protocol

The first two steps of the protocol (4.4) and (4.5) can be combined if required, but in many practical circumstances progressive keying is valuable since it allows the choice of Diversity Keys to be delayed near to the point of use. Importantly, it also allows the choice of the user keys to be deferred; the security of the system depends on protection of the user's private key and it is therefore important to be able to defer the selection of this key, allowing the use of different keys in different theatres of operation and at different times.

The actual formatting of the message in step 4.6 is critical. It is important that an attacker is unable to separate D_j and B_i , because if this is possible then by selecting motes from a range of locations an attacker would be able to use n subverted motes to generate n^2 identities. As noted above (Section 4.1.5), suitable public key components and mes-

sage compilation methods must be used to ensure that the message can only be constructed by a system in possession of the content and the appropriate key.

The final part of this protocol ensures that the user's Comparison Group comprises a set of physically different motes (4.8), that they are situated in approximately the correct area (4.9), and that any that are statistically over represented are rejected (4.10). Section 5 shows that these criteria provide an effective defence against attacks that try to swamp the system with very large numbers of alien devices.

The previous protocol discussed communication threats to the system, now the complete system has been described it is possible to consider attacks on the configuration management process:

After manufacture. An attacker obtains, and perhaps updates, batches of dust motes prior to deployment. This enables the attacker to obtain or set an unencrypted Batch Key B_i . Even if the attacker can obtain devices from the same batch after they have been fielded it would only be possible to find Diversity Keys D_j by trial and error. Provided the Diversity Keys are large enough to resist exhaustion the attacker is unable to generate more Authentication Tokens than can be obtained directly from motes.

Deployment mechanism. An attacker obtains a deployment system containing operational keys. This provides the attacker with a number of Diversity Keys D_j and a user public key E_u . The situation is almost symmetrical with previous case, in that possession of Diversity Keys does not allow an attacker to generate Authentication Tokens without possession of Batch Keys, which in turn need to be large enough to resist an exhaustion attack. However, while Batch Keys are limited to a specific location, Diversity Keys may be used over a much wider area. For this reason, the risk of the loss of a keyed deployment mechanism should be considered carefully in the context of specific deployment risks; in many instances it may be better to separate the keying of a batch of smart dust from its physical deployment to minimise this risk.

Dust mote. An attacker obtains and successfully analyses the content of a dust mote. The attacker obtains a valid Authentication Token, A_m , and the

public key of the user, E_u . This allows the attacker to impersonate the mote and to provide misleading information to a user. The user's defence is to make use of multiple motes with different Diversity Keys; the structure of the Authentication Token prevents the attacker manufacturing multiple Diversity Keys in order to impersonate many different devices so the attacker is limited to impersonating the number of devices that can be analysed from the immediate location of the desired attack.

User. An attacker obtains a user's private key. This is a serious attack on the system, it would allow an attacker to use the Introduction Protocol to obtain Authentication Tokens from motes in place, without the need for physical analysis; simulations of the dust system could then be deployed that could effectively impersonate sensors to mislead other users. Clearly, the user's private key needs to be as well protected as any other master cryptographic key. In some scenarios the degree of risk may mean that these protocols need to be routed to infrastructure that can be physically protected, rather than processed directly in the field.

5. Performance analysis

The Configuration Trust framework proposed in this paper relies on the selection of a number of different sensors and the comparison of their results. The previous section has shown that the protocols used to establish trust are robust against a wide range of standard attacks; this section explores how effective this system is in preventing an attacker from overwhelming a Comparison Group.

The most feasible sophisticated attack on this system is for an attacker to obtain dust motes from an area of interest, analyse them to obtain Authentication Tokens and use those tokens to simulate valid motes which are then used to provide misleading information. An attack of this sort demands significant determination and resources, and will therefore only be attempted if the attacker has a particular outcome in mind. The critical performance metric for this Configuration Trust

system is therefore the attacker's confidence in a successful outcome.

5.1. The analysis scenario

We assume that there are a large number of sensors in a given location and that they originate from a common batch, thus sharing a common Batch Key. (This is the least favourable assumption, since it ignores key diversity due to location). The batch has been primed, at random, from a set of K Diversity Keys, and the user will combine the results from S sensors in such a way that a majority of correct sensors will provide a correct result (see Section 3.6), and a majority of alien sensors will result in a successful attack.

The attacker obtains and analyses dust motes, obtains their Authentication Tokens and the User's Public Key and uses that information to simulate real motes. We assume that it is hard to obtain and analyse sensors from the target location, the sensors will provide some physical protection for their Authentication Tokens, and the attacker may risk detection because of the presence of the sensor network itself. On the other hand, a determined attacker would be capable of using dust technology to try to subvert the sensor network by deploying similar, or even higher, device densities. The key parameters for an attacker are the number of different Authentication Tokens that have been recovered from the location to be attacked (K_a),⁶ and the ratio of subverted devices to real devices, which we shall refer to as the relative density (D). (K_a is likely to be small. $D = 1$ implies that there are equal numbers of authentic and alien devices and therefore a user has the same probability of contacting an alien mote as an authentic device).

The user follows the protocols given above to build a Comparison Group, ensuring that their Batch Keys are consistent (protocol 4.9), discarding motes with the same Diversity Key (protocol 4.8)

⁶ In a single location the number of different Authentication Tokens identifies the number of different Diversity Keys.

and eliminating Authentication Tokens that are observed more than once (protocol 4.10), until a group of S distinct sensors has been assembled. The results from this group are then merged to provide whatever information the user is seeking.

5.2. Theoretical analysis

An analytic model of the formation of valid Comparison Groups is complicated by the dependence of the protocol on other putative members of the group (4.8) including the possibility of removing selections that have already been made (4.10). However, it is possible to identify the controlling parameters in this process, and how they relate to each other.

Consider the formation of a Comparison Group of size S . We are concerned with the likelihood that attacker is able to overwhelm the group, in which case the user must have selected at least $n = 1 + S/2$ alien devices. Despite the many possible combinations of authentic and alien selections, if the probability of selecting n alien devices was constant, it would appear as a common multiple in the probability of each possible combination.

Unfortunately the probability of selecting n alien devices is not constant, since the selection of authentic devices may consume Diversity Keys that further restrict the choice of alien devices. However, if the total number of keys, K , is large with respect to the number of subverted keys, K_a , then few authentic selections will be in the set of K_a keys, so the probability of selecting n alien devices is approximately constant, and can be used to investigate how the system parameters influence the overall likelihood of a successful attack.

This is the basis of the following theoretical analysis, the accuracy of its predictions will be explored quantitatively in Section 5.3.

The probability of a user initially selecting an alien device is simply

$$P(\text{alien}_0) = \frac{D}{1 + D}. \quad (5.1)$$

The total number of devices does not appear in the following equations, they are normalised to the number of Diversity Keys in use. This assumes

that the number of authentic notes are distributed uniformly between the available keys and that the optimum strategy for an attacker is similarly to spread recovered keys evenly between alien devices.

After n devices of any kind have been selected, the remaining number of keys available for selection (which is the same as the normalised number of available authentic notes) is

$$\text{Total}(\text{authentic}|n) = K - n. \quad (5.2)$$

If the user has already selected n alien devices, the only devices available for selection are those with different Diversity Keys (see protocol 4.8), so the normalised number of available alien notes is

$$\text{Total}(\text{alien}|n_{\text{alien}}) \approx DK \frac{(K_a - n)}{K_a}. \quad (5.3)$$

The probability of selecting an alien sensor, following the selection of n previous alien sensors is therefore

$$P(\text{alien}|n_{\text{alien}}) \approx \frac{DK \frac{(K_a - n)}{K_a}}{DK \frac{(K_a - n)}{K_a} + (K - n)} \quad (5.4)$$

or

$$P(\text{alien}|n_{\text{alien}}) \approx \frac{D}{D + \frac{K_a}{K_a - n} \cdot \frac{K - n}{K}}. \quad (5.5)$$

These equations give a clear view of the controlling parameters in this process. The major factor in this probability is the attack density, D . An attacker will be forced to deploy similar, or higher, device densities to have any confidence in the success of an attack.

If K is large with respect to K_a , then as n increases the term $(K - n)/K$ remains close to unity, but the term $K_a/(K_a - n)$ increases quickly, rapidly decreasing the probability that an attack device will be selected. The total key size, K , therefore has little effect on the performance of the system, provided it remains large with respect to the number of keys that can be compromised.

This is an important observation, since if K is large enough, then each element of dust has its own identity; the implication is that providing each element with its own identity has very little

impact on the robustness of the system against attacks of this sort.

The speed at which $K_a/(K_a - n)$ grows depends upon the relationship between n and K_a ; of course, the values of n of interest are those near $(1 + S/2)$, the point at which an attacker has obtained a majority in the Comparison Group. The relationship between K_a and S , the size of the Comparison Group, is therefore critical; intuitively we would expect that if an attacker is able to subvert many more keys than S , then the chance of a successful attack would be higher; however, the numerical results below demonstrate that although the performance of this trust system does degrade when $K_a > S$, it does so gracefully rather than catastrophically.

5.3. Quantitative analysis

The previous section uses approximate probability functions as a basis for performance prediction, since obtaining an analytic solution is not straightforward. In order to confirm these predictions, this section presents a quantitative evaluation of the system, obtained using Monte Carlo methods.

A practical system would set $S/2$ greater than the estimated capability of the attacker, K_a . If this estimate was correct, the system could never be attacked. However, the following will investigate how robust this system is if the attacker’s capability is underestimated, by using a range of values for S and K_a .

5.3.1. The impact of high attack densities

The major factor in this analysis is the relative density of alien devices deployed by an attacker; Fig. 4 provides a straightforward evaluation of the probability of a successful attack using the simple probability model above (the ‘without filtering’ result) for a range of relative densities. (Other parameters used are $K = 50$, $K_a = 7$, $S = 7$).

The underlying factor of $D/(1 + D)$ dominates this result, although the requirement for a selection to include different Diversity Keys (protocol 4.8) reduces the attack probability from the factor 0.5 that would otherwise be expected when the $D = 1$,

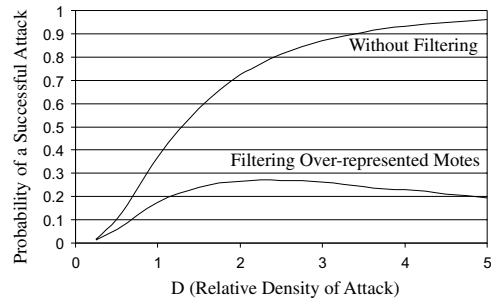


Fig. 4. The dominance of attack density in the simple probability model, and its management by filtering over-represented motes (protocol 4.10).

to a value of 0.37, with similar reductions at higher densities.

The protocol, however, has a requirement (4.10) that is not included in this simple calculation, and is designed to deal with high density attacks.

The probability of communicating with an authentic device, with a given Diversity Key, k , is

$$P_{\text{communicate}}(k \cdot \text{authentic}) = \frac{1}{K(1 + D)}. \quad (5.6)$$

However, an attacker is forced to concentrate the available motes on only K_a keys, so the probability that communication is attempted with an alien device where k is within the set of K_a subverted keys, is

$$P_{\text{communicate}}(k \cdot \text{alien} | k \cdot \text{within } K_a) = \frac{DK}{K_a} \cdot \frac{1}{K(1 + D)} = \frac{D}{K_a(1 + D)}. \quad (5.7)$$

Using the parameters from Fig. 4 ($D = 5$, $K = 50$, $K_a = 7$), a random communication is 36 times more likely to select a subverted key. Variations of this order are rather easy to identify statistically, even with small numbers of samples. There are a number of ways of estimating which keys have been subverted, ranging from a centralised infrastructure shared by many users, to individual user histories, or the simple stateless approach reflected in protocol step 4.10. The approach proposed in the protocol has the benefit of being stateless, scenario independent, and from the point of view of this analysis, conservative.

Protocol step 4.10 removes from a Comparison Group any Authentication Tokens that are

observed more than once during its formation, in order to avoid over-represented notes. Provided that an attacker is able to subvert only a small number of keys, then deploying higher densities of alien devices in order to increase the likelihood that they are used also increases the probability that they are detected and excluded. The performance of the protocol with this feature is given in Fig. 4 for comparison (‘with filtering’).

This is an encouraging result, since it indicates the possibility of designing dust-based Configuration Trust systems that are extremely difficult to overwhelm.

5.3.2. The relationship of S to K_a

An important factor in the performance of this protocol is the relationship of S to K_a . Intuitively, it seems likely that the user of this system should select the size of the Comparison Group S on the basis of a risk assessment of the capability of an attacker to obtain keys. Fig. 5 illustrates the performance of the protocol for an arbitrary value of K_a (7) and a range of choices for S .

Increasing the value of S relative to K_a reduces the likelihood that there sufficient subverted keys available to overwhelm a Comparison Group (of course, if $K_a < S/2$, then the attacker never succeeds in overwhelming a user). Increasing S also increases the probability that in the course of searching for S sensors then subverted Authentication Tokens are identified and eliminated. This double factor is illustrated by the degree to which small increases in S produce large increases in performance.

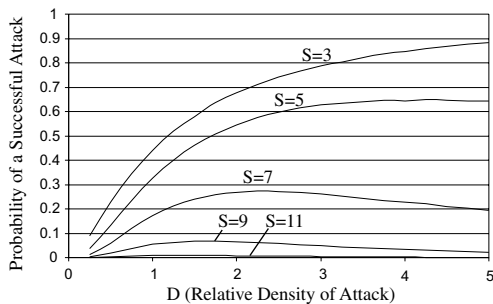


Fig. 5. S and K_a , the probability of a successful attack for a range of values of S .

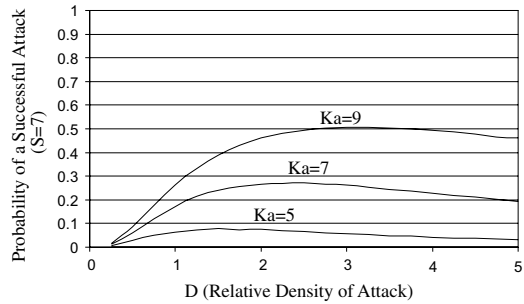


Fig. 6. S and K_a , The Attackers Perspective. The probability of a successful attack for a range of values of K_a .

Fig. 6 provides an alternative view of this process by selecting a constant S , and investigating the work factor from the attacker’s perspective.

The critical criterion for the success of this Configuration Trust system is the extent that it denies the attacker confidence that an attack will succeed. This result illustrates the strength of the system from that perspective. $S = 7$ is a realistic and small Comparison Group, $K_a = 9$ is a large proportion of the available keys to be compromised. Deploying large numbers of alien motes fails to provide the attacker with a high confidence, and key rejection within the protocol will alert the user to the attack.

We believe that this is a powerful demonstration of how robust this protocol is in the face of a determined and resourceful attack.

5.3.3. Number of Diversity Keys

The final observation made in the theoretical analysis, was that the number of Diversity Keys deployed in the system has little effect on its performance. Fig. 7 repeats two of the family of

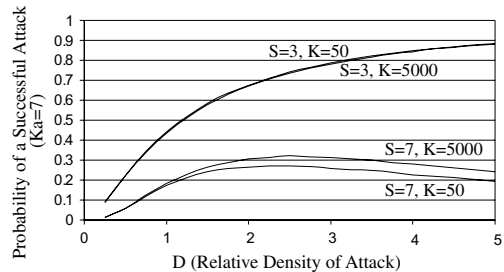


Fig. 7. Impact of the size of the Diversity Key set on system performance.

results from Fig. 5 using the original number of Diversity Keys (50) and a much larger number (5000) for comparison.

These results confirm the observation that the number of Diversity Keys has little effect on the performance of the system. The density of the attack and number of keys that are compromised with respect to the size of the Comparison Group are much more significant factors. This suggests that for most practical purposes a small number of Diversity Keys would suffice, and confirms that providing each element of dust with its own identity does not provide additional defence against this form of attack.

6. Simulation results

Simulations of dust deployments are scenario specific, but the random physical element in deployment and the difficulty of obtaining an analytic expression for the process probability, mean that simulation is necessary in order to investigate the effectiveness of location correlation using Batch Keys and to study the variance of the likelihood of success of an attack between different physical deployments. This section provides initial results from a dust deployment simulator.

Possible physical deployment mechanisms include point, linear and area distribution. Point distribution essentially corresponds to the analysis above: the Batch Key is assumed to be a constant for all the sensors in that location. Linear and area distribution may be drawn from a series of batches so the effectiveness of Batch Keys may be investigated in either of these cases. Areas may be represented as a series of linear tracks, so the building block of the simulator is a linear track.

In the trial presented here, the model for dust distribution is that dust is released progressively along the length of the track, with uniform random angle and a uniform random velocity that is chosen to provide a given track width. This is similar in effect to a spinning distributor in an aircraft. The model allows the Batch Key to be changed at intervals as the dust is distributed.

An attack is also simulated as the distribution of dust motes; the attacker selects K_a dust motes

already present in an *attack corridor* across the track, and distributes alien dust using these captured Authentication Tokens uniformly in the corridor.

The simulated user chooses S items of dust at random from within the corridor, ensuring that they have different Diversity Keys.

The user also ensures that they have a plausible Batch Key. In this trial the simulator checks that the linear track distance to the centre point of distribution of that batch is not greater than could have been achieved by the distribution mechanism. It is outside the scope of the paper to deal with location strategies in detail, but this is more conservative (i.e. less effective) than the simple strategy of ensuring that all members of a configuration group have the same Batch Key. The previous results did not measure the effect of Batch Keys, the purpose of simulating a weaker strategy than that proposed in the protocol is to enable comparison with these results.

Fig. 8 provides a pictorial image of a simulated track, the middle half of this track is the attack corridor, and the solid marks are alien devices, with the same overall density as dust sensors; the changing symbols marking authentic dust indicate different Batch Keys, in this trial the Batch Keys are changed at distances corresponding to the overall track width.

It can be seen from this track that the attack corridor is mostly associated with a single batch of devices, although drift from adjacent batches overlaps. This layout has been used in the simulation, since it presents the most favourable position for an attacker.

Fig. 9 provides a family of histograms of the distribution of the probability that an attack

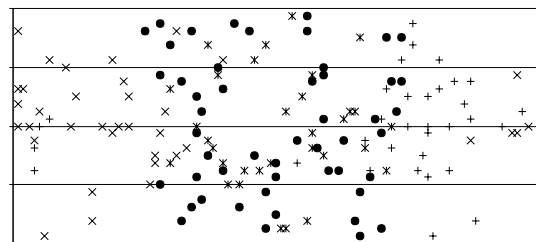


Fig. 8. An image of a typical track, at low density.

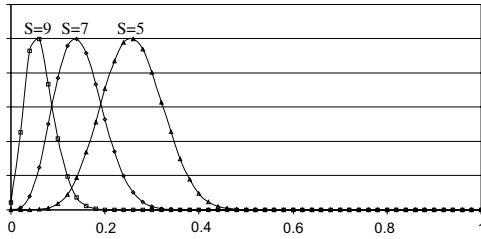


Fig. 9. Distribution of the likelihood of a successful attack for a range of values of S .

succeeds, and should be comparable with the results in Fig. 3 at a density of unity ($K = 50$, $K_a = 7$). Each trace in the figure presents a simulation trial averaged over 10^6 different deployments, with 100 attempts to form a Comparison Group in each deployment.

The means of each these distributions are slightly better than those predicted in Fig. 3 (e.g. for $S = 5$, the predicted mean is 0.32, whereas in the simulation it is 0.25). This difference is accounted for by location filtering; on average, these trials rejected 8% of the keys tested because they were not in valid locations, despite the approximate alignment of the attack corridor with a batch and the conservative location criterion.

It is not appropriate to draw detailed conclusions from a limited and scenario specific simulation, but it is encouraging to note the close correspondence between these results and those predicted in the analysis, that the measured variance is sufficiently narrow for the approach to be workable in practice, and that location filtering has the prospect of providing further improvements to the performance of the Configuration Trust system.

7. Further work

This paper has provided a complete discussion of the topic of Configuration Trust, including requirements, protocols, analysis and simulation results. Inevitably, there is scope for further development in each of these areas.

Location filtering and the use of Batch Keys force an attacker to work in place. The additional robustness that accrues from filtering batch keys

(protocol 4.9) has not been quantified in this paper, but the simulation demonstrates that it has the prospect of further improving the performance of this system. Other researchers have demonstrated similar benefits in other key distribution schemes [25].

This protocol includes a stateless mechanism for identifying subverted keys; it seems likely that a statistical criterion can be designed that uses the experience of an individual user without incurring additional infrastructure. This diverges from our objective to develop a stateless protocol, but may further improve the quality of attack detection.

Finally, an important future topic is how to achieve Configuration Trust in other scenarios. Ubiquitous systems in general will be able to form ad hoc networks; the ideas presented here for the dust scenario may be useful in showing how to deconstruct the authentication problem in other circumstances.

8. Conclusion

This paper has identified an important problem in ubiquitous computing, *Configuration Trust*, which is concerned with how a user can trust the behaviour of a service provided by a network subsystem. Designing with this in mind provides a new perspective on the security of highly distributed systems.

This perspective has been used to study the security of Smart Dust systems in a military environment; this problem has physical properties that differentiate it from other ubiquitous system research, and a threat environment that stresses the importance of integrity rather than confidentiality.

The paper demonstrates a viable alternative to a conventional infrastructure where each device has its own identity. In this system the user builds trust in the integrity of sensor outputs by exploiting the server rich environment. At any instant a user fuses the results from a group of sensors; the group is chosen to ensure batch consistency, which correlates to consistent location, and diversity, which prevents an attacker overwhelming the group with subverted devices.

An important result (see Section 5.3.3) is the comparison between this new trust process and a conventional identity-based system. This process offers comparable performance, without the overhead of supporting identities for individual nodes.

This process is stateless, ensuring that a user does not need the support of history or infrastructure to make decisions, and also robust, in that it is able to deal gracefully with attacks using overwhelming numbers of devices.

The work described in this paper provides a complete coverage of the problem, dealing with requirements, design approaches, supporting protocols and performance analysis, and shows that these results are consistent with those obtained in a simulated ‘dust war’.

Acknowledgements

The authors thank the anonymous reviewers for their helpful and constructive comments on this paper.

References

- [1] K. Pister, On the limits and applications of MEMS sensor networks (2001), Defense Science Study Group Report, Institute for Defense Analysis, Alexandria, VA, 2001. Available from <<http://citeseer.nj.nec.com/pister01limits.html>>.
- [2] T.-H. Lin et al., Wireless integrated network sensors (WINS) for tactical information systems, in: 1998 Government Microcircuit Applications Conference, 1998. Available from the WINS project, at: <http://www.janet.ucla.edu/WINS/biblio.htm>.
- [3] J.M. Kahn, R.H. Katz, K.S.J. Pister, Next century challenges: mobile networking for “Smart Dust”, in: International Conference on Mobile Computing and Networking (MOBICOM), ACM, 1999, pp. 271–278.
- [4] J. Hill et al., System architecture directions for networked sensors, in: Ninth International Conference on Architectural Support for Programming Languages and Operating Systems, ACM Press, 2000, pp. 93–104.
- [5] T. Clouqueur et al., Sensor deployment strategy for target detection, in: The 1st ACM International Workshop on Wireless Sensor Networks and Applications, ACM, 2002.
- [6] Q. Hairong, S.S. Iyengar, K. Chakrabarty, Distributed sensor networks—a review of recent research, The Mathematics Preprint Server, 2001. Available from <<http://www.mathpreprints.com/math/Preprint/hqi/20010706/1/?=&coll=Selection>>.
- [7] The Common Criteria, 1999, Common Criteria Support Environment (CCSE). Available from <<http://www.commoncriteria.org/cc/cc.html>>.
- [8] H. Chivers, J.A. Clark, S. Stepney, Smart devices and software agents: the basics of good behaviour, in: The First International Conference on Security in Pervasive Computing, Springer, Berlin, 2003, pp. 39–52.
- [9] B. Schneier, Applied Cryptography, second ed., Wiley, New York, 1996.
- [10] M. Gasser, E. McDermott, An architecture for practical delegation in a distributed system, in: IEEE Symposium on Research in Security and Privacy, IEEE, 1990, pp. 20–30.
- [11] Web Services Trust Language (WS-Trust), 2002, IBM.
- [12] L. Bussard, Y. Roudier, Embedding distance-bounding protocols within intuitive interactions, in: Proceedings of the First International Conference on Security in Pervasive Computing (SPC2003), Springer, Boppard, Germany, 2003.
- [13] Y.-C. Hu, A. Perrig, D.B. Johnson, Packet leases: a defence against wormhole attacks in wireless networks, in: IEEE INFOCOM 2003, IEEE, 2003.
- [14] N. Sastry, U. Shankar, D. Wagner, Secure verification of location claims, in: 2003 ACM Workshop on Wireless Security, ACM Press, 2003, pp. 1–10.
- [15] B. Shand, N. Dimmock, J. Bacon, Trust for ubiquitous transparent collaboration, in: The First IEEE International Conference on Pervasive Computing and Communications (PerCom’03), IEEE, 2003, pp. 153–160.
- [16] A. Abdul-Rahman, S. Hailes, A distributed trust model, in: 1997 Workshop on New Security Paradigms, ACM Press, 1997, pp. 48–60.
- [17] V. Cahill et al., Using trust for secure collaboration in uncertain environments, IEEE Pervasive Computing 2 (3) (2003) 52–61.
- [18] F. Azzedin, M. Maheswaran, Trust modeling for peer-to-peer based computing systems, in: The International Parallel and Distributed Processing Symposium (IP-DPS’03), IEEE Computer Society, 2003.
- [19] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: 9th ACM Conference on Computer and Communications Security, ACM, 2002, pp. 41–47.
- [20] R. Blom, An optimal class of symmetric key generation systems, in: EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques, Springer, Berlin, 1985, pp. 335–338.
- [21] C. Blundo et al., Perfectly-secure key distribution for dynamic conferences, in: CRYPTO ’92: 12th Annual International Cryptology Conference, Springer, Berlin, 1992, p. 471.
- [22] D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: 10th ACM Conference on Computer and Communication Security, ACM Press, 2003, pp. 52–61.
- [23] W. Du et al., A pairwise key pre-distribution scheme for wireless sensor networks, in: 10th ACM Conference on

Computer and Communication Security, ACM Press, 2003, pp. 42–51.

- [24] H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, in: 2003 IEEE Symposium on Research in Security and Privacy, 2003, pp. 197–215.
- [25] W. Du et al., A key management scheme for wireless sensor networks using deployment knowledge, in: IEEE InfoCom 2004, IEEE, 2004.
- [26] W. Du et al., A witness-based approach for data fusion assurance in wireless sensor networks, in: IEEE 2003 Global Communications Conference (GLOBECOM), IEEE, 2003.
- [27] A. Perrig et al., SPINS: security protocols for sensor networks, in: The Seventh International Conference on Mobile Computing and Networking, ACM, 2001, pp. 189–199.
- [28] M. Beller, Y. Yacobi, Fully-fledged two-way public key authentication and key agreement for low-cost terminals, *Electronics Letters* 29 (11) (1993) 999–1001.
- [29] P.K. Varshney, *Distributed Detection and Data Fusion*, Springer, New York, 1996.
- [30] R.L. Rivest, A. Shamir, L.M. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM* 21 (2) (1978) 120–126.



Howard Chivers B.Eng, C.Eng, F.I.E.E. is a Royal Academy of Engineering Research Fellow at the University of York. His career in Industry and Government has included cryptography, communications, and the design of large secure data processing systems. From 1987–1991 he managed the computer systems security research program for CESG. Current research interests include design approaches to security in highly distributed systems, including grid, web services and pervasive computing environments.



John A. Clark M.A., M.Sc., Ph.D. is Senior Lecturer in Critical Systems at the University of York. Before joining York in 1992 he worked on UK Government-funded evaluation and R&D security projects. He has provided consultancy to industry on various aspects of dependability modelling. Current research includes software testing, security patterns, anonymity, nature-inspired approaches to dependability and the evolutionary synthesis of cryptographic elements, protocols and quantum algorithms.