Identity Crisis: On the Problem of Namespace Design for ID-PKC and MANETs

Shane Balfe*, Andrew D. McDonald†, Kenneth G. Paterson* and Helen Phillips §

*Royal Holloway, University of London, Egham, Surrey, TW20 0EX, UK. – Email: {s.balfe,kenny.paterson}@rhul.ac.uk
†Roke Manor Research Ltd, Romsey, Hampshire, SO51 0ZN, UK. – Email: andrew.mcdonald@roke.co.uk
§Defence Science and Technology Laboratory, Malvern, Worcester WR14 3PS, UK. – Email: hlphillips@dstl.gov.uk

Abstract—In this paper we explore the “interface” between Identity-based Public Key Cryptography (ID-PKC) and Mobile Ad-hoc Networks (MANETs). In particular we examine the problem of naming and namespace design in an Identity-based Key Infrastructure (IKI). We examine the potential impact that different types of identifiers may have on the utility of ad hoc networks where an IKI provides the underlying key infrastructure. We also highlight problems inherent in extending namespaces to allow inter-operability amongst heterogeneous trust domains.

I. INTRODUCTION

Recently, the advent of practical identity-based public key cryptographic schemes has resulted in a number of proposals for using Identity-based Public Key Cryptography (ID-PKC) as the underlying key management infrastructure for Mobile Ad-hoc Networks (MANETs) [24], [30], [44]. Here the use of ID-PKC promises lower bandwidth and energy requirements over traditional PKIs, the capability for non-interactive key establishment, and the potential for better scalability of user registration and key management processes [8]. However, the use of ID-PKC in MANETs necessitates re-examining the fundamental building block of an Identity-based Key Infrastructure (IKI) [17], that of identity.

The interplay between identity and key management is perhaps the most complex and most vulnerable part of any cryptographic implementation. It is vitally important that everything possible is done to develop careful procedures for managing keys, and the identities to which these keys are bound. In a Public Key Infrastructure (PKI) [26], identity management is handled by a Certificate Authority (CA) binding a subject’s identity to a public key in the form of a public key (identity) certificate. A relying party, in establishing a subject’s identity is completely dependent on the CA’s enrollment procedures if it is to have faith that an identity claimed by an enrolled subject is the actual claimant’s identity.

By contrast, in an IKI, there is no verifiable third party binding between an identity and a public key. In an IKI, identities are public keys. A relying party wishing to send an encrypted message to a subject can do so knowing only the identifier string of that subject, along with the public parameters of the Trusted Authority (TA) to which that subject belongs (see Section II-A). This poses two problems for any IKI solution. Firstly, a subject must be able to demonstrably prove ownership of a claimed identity when interacting with a TA to obtain the corresponding private key. Secondly, unless identities are pre-established and well-known, a relying party must be either reliably informed of a subject’s identity or choose an identity for a subject with which it wishes to communicate.

Effective identity management poses one of the greatest challenges to delivering an IKI for MANETs. The enrollment of subjects (and the provisioning of cryptographic keys) within an Identity Management System (IMS) encompasses the creation, maintenance and termination of a subject’s identity. Traditional approaches to identity management such as the identity silo (a closed community of users within one IMS), the walled garden (one IMS serving a number of organisations) and identity federation (multiple autonomous IMSs that inter-operate based on a-priori trust relationships) fail to adequately capture the dynamism of ad-hoc networks. The spontaneity with which MANETs may be formed, the frequency with which individual nodes may join or leave a network, and the possibility of multiple networks dynamically composing all make static pre-established trust relationships undesirable. The design of an identity management infrastructure that can adapt to the dynamic nature of MANETs, whilst being mindful of their unique constraints, poses many enigmatic problems.

This paper sets out to examine one particular problem with the design of an IMS (with an IKI as the underlying key management infrastructure) for MANETs, that of naming and namespace design. Naming provides a means of mapping a name (identity) to an object and is typically used to distinguish one object from another. In [32], Needham, in highlighting some subtleties of naming, draws a distinction between pure and impure names [32]. A pure name can simply be interpreted as an identifier or pointer, and is only useful in testing for equality with other identifiers. By contrast, impure names incorporate additional information and require a context in

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Government or the U.K. Ministry of Defence. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.
which the name will be interpreted. Namespaces can be viewed as containers for providing contexts for (impure) names (identities), allowing subjects having the same name (but residing in different namespaces) to be distinguished.

In this paper we explore the “interface” between ID-PKC and MANETs and we examine the potential impact that different types of identifiers may have on the utility of ad hoc networks. We also examine the issues that arise in namespace design for facilitating the description of identities and relationships within a given domain. Finally, we highlight problems inherent in extending namespaces to allow inter-operability amongst heterogeneous trust domains, each of which may have their own separate namespace. This paper does not aim to provide detailed solutions to these problems but aspires to provide the first steps in exploring the current problems faced in the design of namespaces for an Identity-based Key Infrastructure.

This paper is organised as follows. In Section II, we provide a definition of identity and examine how identity has been captured in secure communication systems. In Section III, we look at how namespaces are constructed for categorising identifiers. In Section IV, we look at how identity is used at different layers of a protocol stack, examine relationships between identities at different layers, and note how identity impacts on the correct functioning of the network, as well as on security. In Section V, we look at attacks that can be launched against identity and briefly examine some defences to these attacks. Finally, we conclude with Section VI.

II. WHAT IS IDENTITY?

Identity can be viewed as the ability to distinguish one subject from another (in a given context) and forms a cornerstone of how one reasons about the world. At birth\(^1\) a subject is given a name by their parent(s), and it is through this name that other subjects come to identify this subject. However, irrespective of this name, a subject may later choose to adopt a new name for themselves. A subject can simply "earn" their new name through use and repute. For most purposes, if a subject wishes to be known by an alias, they simply use this new name and ask other subjects to call them by it. Here identity is a claim made by a subject that a name belongs to them and can be used to identify them within a given context. For the remainder of this paper, we adopt the definition presented in [11], identity is a “set of claims made by one digital subject about itself.” These claims may be an assertion that a subject knows a given key, that a particular identifier ‘belongs’ to them or that the subject is part of a certain group.

A. The Use Of Identity In Cryptography

In 1976 the cryptographic landscape changed when Diffie and Hellman postulated that the key management problem could be solved through the introduction of a centralised authority, the Public File. The Public File (much like a modified telephone directory) would allow a user to perform a lookup based on some identifying information to obtain another user’s public key. However, this concept of a directory in which an identity could be looked up was deemed impractical for networked environments (as it introduced single points of vulnerability/failure). This quixotic notion of a permanently available central repository led to Kohnfelder’s proposal of public key certificates [29]. Here the the Public File would be split into individual tuples which could be digitally signed by a CA, and freely distributed. In this way a peer could assert their identity by distributing their certificate instead of relying on a communicating peer performing a lookup on a centralised directory. This later led to standardisation efforts in the form of the X.509 certificate format [25] for expressing the binding between a public key and a subject’s identity.

However, in addressing the key management problem public-key cryptography has exacerbated a series of inter-related problems of managing identities [20]. How does the CA determine the truthfulness of A’s claimed identity prior to certification? What happens if A’s private key were to become compromised? And from where will B obtain a copy of A’s certificate? In addition to these questions there is the issue of establishing a subject’s identity with an entity from a foreign CA. Establishing a relationship between the domain in which the entity is enrolled and the domain to which the entity wishes to authenticate is typically achieved through cross certification. However, in doing so issues such as re-parenting, subordination of one CA to another, revocation and re-issuance/replacement, and the hierarchy of trust all become major hurdles that need to be addressed. Different CAs and paths may have different validity periods and constraints; certificate paths can contain loops and certificate semantics can change on different iterations through the loop [23]. Issues such as these are typically not handled in the MANET literature. Instead these questions are left to some undefined and largely unexamined policy “layer”.

B. Identity-based Cryptography

In 1984, Shamir introduced the concept of identity based cryptography [37]. However, it took nearly twenty years for an efficient and provably secure Identity Based Encryption (IBE) scheme to be proposed [10]. By allowing public keys to be derived from a combination of public system parameters and information that uniquely identifies an subject, such as an email address, ID-PKC obviates the need for identity certificates. However, the use of ID-PKC is not without its drawbacks. A summary of these can be found in [34], [28]. The identifying information that a sender uses to create a public key for a given recipient may belong to the wrong recipient, or be considered invalid by a TA when deriving the recipient’s private key, rendering the message unreadable. The use of identities in ID-PKC makes the authenticity of identifying information a crucial component in the system, as

---

\(^1\) Here birth can be interpreted as the physical birth of a human or the initialisation of a device as per [41].
public keys may be used prior to the corresponding private key being generated.

As a corollary, to use of an subject’s identity in the derivation of a public key, key revocation becomes somewhat more problematic as the public key is, in essence, an subject’s identity. To address these issues, additional information can be encoded in deriving a public key that offers a circumlocutory way of expressing subject identity by including validity dates, key usage restrictions and other pertinent information. However, such information makes determining a subject’s identity more difficult as a subject’s identity is now an amalgam of name and policy information.

Recent advances in ID-PKC have led to Hierarchical Identity-Based Encryption (HIBE) being proposed. In HIBE schemes, naming follows a hierarchy, where an identifier is defined via the path from a root down to a leaf node. As we will see in Section III, this may provide a natural mechanism for describing command and control hierarchies but comes at a cost of reduced privacy.

A relatively new identity-based encryption scheme has been proposed that uses wildcards as part of the identifier space [1] allowing an efficient and secure form of broadcasting. In a Wildcard Identity Based Encryption (WIBE) scheme, encrypted messages can be addresses to a group of entities, any one of which is able to decrypt to obtain the plaintext. The group is defined in a hierarchical namespace by allowing wildcards at certain levels in the namespace. Such a scheme could be used to send a secure query to all nodes at the lowest level of an IBE hierarchy, without first having to learn the identifiers of all the local nodes.

III. Namespaces – What’s In A Name?

The security of an IKI for MANETS is predicated on the availability of an efficient and effective naming (identity) infrastructure. However, the challenges presented by identity management in MANETs makes traditional notions of public-key infrastructures unwieldy. Schemes based on identity silos, walled gardens and federation may be impractical for networks comprised of highly mobile nodes as limitations on communication ranges for individual nodes, unknown network topologies prior to deployment and intermittent node operation may render such approaches ineffective.

A. Global Versus Local Namespaces

A global namespace attempts to uniquely identify an individual amongst all individuals. An example would be a user’s X.500 distinguished name in an X.509 certificate. The problem with a global naming scheme, however, is that multiple users may have homonymous names, leading to cases of mistaken identity. Within a single statically-scoped domain, there may be two entities with names that are either identical or phonetically equivalent, thus potentially indistinguishable to a human observer. This raises the question: how can an outsider determine which entity it should be communicating with? In an attempt to address this question the idea of linked-local namespaces has been proposed in the literature [21]. In a local namespace, a user defines how they refer to other entities themselves. Here, instead of relying on some global naming scheme, names have local significance only. It does not matter if two people call different entities by the same name. What matters is that, within the namespace that each user creates for themselves, names be unique. An example of such a system is the Simple Distributed Security Infrastructure (SDSI) naming scheme used in Simple Public key infrastructure (SPKI) [21]. SDSI, like X.509, is predominately concerned with the binding of identities to public keys. In SDSI, active agents within the system are public verification keys. Users assign names to public keys as shorthand for identifying the key’s owner. These namespaces are then exportable by an entity by signing their namespace (or a portion thereof) and distributing it to other entities. In this way, namespaces may be linked, and one principal may use the namespaces defined by another. For example, if principal Alice knows principal Bob, and Bob has exported a portion of his namespace where he defines principal Eve, principal Alice may refer to principal Eve as Bob’s Eve. Whilst this approach addresses the problem of distinguishability in a global namespace it introduces some extra problems in the case of ad hoc networking. If we were to adopt an unmodified SDSI implementation as the naming infrastructure for an IKI, there could be an explosion in the number of identifiers for each principal (node). This is because each principal could be referred to in multiple ways by multiple other principals, building arbitrarily long chains of references as identifiers.

SDSI works well for traditional PKIs as it is possible to tell if two individuals are referring to the same entity in their separate namespaces. This is possible as principals are their public keys and public keys can be tested for equivalence. The same is not true for ID-PKC as public keys are derived from identities and not the other way around, as is the case in SDSI. The number of permutations possible to refer to a single entity may become enormous and there is no easy way of telling if two identifiers correspond to the same individual. This could have the resulting outcome of an entity having to make continuous requests (in the process, establishing secure and authentic channels) to its TA in order to obtain private keys for each of the identifiers that other entities could possibly know him by. This problem is compounded in scenarios where the TA role is split over multiple entities to reduce the possibility of TA compromise.

B. Hierarchical Versus Flat Namespaces

Perhaps the two best known hierarchical schemes are the Domain Name System (DNS) and the X.500 namespace used in X.509 certificates. Hierarchical namespaces partition names into progressively smaller components, until an object can be uniquely identified. Hierarchical namespaces rely on a general agreement about the roots of the hierarchy, the names of which should be similarly interpreted in all namespaces. In the X.500 namespace scheme, an entity is unambiguously identified by a Distinguished Name (DN). A DN is the concatenation of selected attributes from the root element all the way down
to the named entity. Path names in file systems follow the common hierarchical pattern of ownership/control and so make a good model for namespace organisation [2]. However, obvious choices for identifiers may not always be desirable. For example, in a military MANET, a natural choice for identity may be to embed one’s identifier in a command hierarchy. However, in doing so we disclose a node’s rank, potentially making that node more vulnerable to attack. Additionally, in order to address a node in an ID-PKC hierarchy, a sender would need to have a global understanding of the hierarchy to construct a node identity.

Flat namespaces, by contrast, contain no additional useful information embedded in the identifier. Recent research in peer-to-peer systems has resulted in scalable and highly distributed flat namespaces organised by Distributed Hash Tables (DHTs) [38]. In a DHT setting, a keyspace is defined that partitions keys (locators of files) amongst n nodes according to a keyspace partitioning scheme. Lookups for keys are performed by routing queries towards the node that has ultimate responsibility for that key. Taking the example of a distributed filesystem, each node that participates in the DHT is assigned a unique ID. Every file in the system is represented by a tuple (key, data) where key is a hash of a filename and data is the binary file. Each ID is then assigned a portion of the keyspace and therefore, by association the filesystem for which the file’s key is closest to the owner’s ID, measured according to an algorithm-specific distance function.

Flat namespaces can be seen to work well if an individual knows in advance the exact identity of the entity/object for which they are searching. However, these namespaces tend to be very application specific.

C. Inter-operability Between Namespaces

In order for entities from one domain to be able to identify entities from another domain, there needs to be some method for bridging different namespaces. This could be achieved through either enforcing a global common namespace across all domains (as is done with X.500) or to introduce some extra infrastructure in the form of a name translation service. Such translation services can be seen in protocols such as the Host Identity Protocol (HIP) [31], the Unmanaged Internet Protocol (UIP) [22] and TurfNet [35].

TurfNet provides an interesting example of how different namespaces may be bridged. Turfs are autonomous networks that may use different schemes for addressing, routing and naming. Naming in TurfNet relies on translation services which map host identities to network layer identifiers for routing and forwarding of packets. Each turf has a gateway that provides a protocol and locator translation service between autonomous turfs. However, such additional infrastructure comes at a price. In a MANET, any additional infrastructural components may need to be distributed amongst the MANET nodes to mitigate the effects of compromise and to add redundancy to the service. This increases the overhead associated with bridging domains.

Issues of inter-operability between different ID-PKC domains have largely been ignored in the literature. Thus far, there have been two architectural proposals for inter-operability of ID-PKC domains, neither of which examines inter-operability of identifiers. Each architecture is primarily concerned with distributing a TA’s public system parameters, as it is these public parameters in conjunction with an entity’s identity, that make up an entity’s public key. In the first architecture, all domains share the same public parameters [15]. In the second architecture, a PKI sitting above differing ID-PKC domains, certifies the ID-PKC’s public system parameters of each domain. In this way, authenticated copies of a domain’s public system parameters can be obtained via a trusted CA at the domain level [13]. In an adjunct to this approach, an ID-PKC domain’s public parameters may be distributed via some authentic channel, such as Secure DNS [40]. However, it remains an open question as to the applicability of these architectures in a MANET setting.

IV. Identity At Different Layers

It is unlikely to be the case that an IMS is going to be implemented from scratch. The more likely scenario is that an IMS will need to integrate into an environment where digital identities are already in place. A real network consists of layers of protocols, using different namespaces at different layers. These namespaces have different requirements and properties depending on the particular protocol they are used with. The names at lower layers (e.g. the link layer) are usually more fundamentally tied to a physical device. At higher layers the concepts of a ‘user’ or ‘resource’ become more significant. These names build on top of (and are translated between) one another, and the relationships between the layers of identity are important for the behaviour of the network, as well as having implications for security. In order to consider the practical issues of identity in a network, this section takes a typical Internet 4-layer protocol stack and examines the identities that are found there. We examine the implications such identities have for MANETs at the end of this section.

A. Link Layer

The IEEE 802 protocols provide the most common form of link layer connectivity, also referred to as the Medium Access Control (MAC) layer. Usually MAC addresses are burnt into hardware, such as network cards, when they are created although it is usually possible to change them in software. The principle requirement on these identifiers is that they are unique on the link. However, these identifiers are actually globally unique. In order to avoid duplicate addresses, assignment is performed by allocating subsets of the identifier space to particular manufacturers, each manufacturer being assigned an Organizationally Unique Identifier (OUI) which then forms part of the Extended Unique Identifier EUI-48/EUI-64. This structure is ignored once the identifier is in use, when the namespace is treated as being totally flat.
B. Network Layer

The network layer provides connectivity across multiple concatenated links and forms the predominant focus of much of the MANET literature. This is the layer at which the concept of global identity, in the form of IP addresses, first become important to the functioning of the protocols. The most ubiquitous network protocol is IPv4, which uses 32-bit addresses to identify hosts and provide the necessary information to route and forward packets to them across the network. IPv6, the successor to IPv4, uses 128-bit addresses in a similar way, although the larger address space allows for some more novel address (identifier) constructions.

IP addresses are allocated in a hierarchical manner to ensure global uniqueness and also provide greater efficiency in constructing routing tables by allowing the aggregation of sets of addresses into single routing rules. This aggregation is less relevant within a MANET, where per-host routing is likely to dominate, but becomes more relevant as network size increases.

However, the architectural assumption of uniqueness may be broken through the use of Network Address Translation (NAT). In this case a ‘local’ IP address may be mapped to a ‘global’ address at the edge of the network, and so many devices can be hidden behind a single address. In this case the ‘unique identifier’ typically incorporates additional higher-layer information (such as transport layer port numbers) to ensure that the NAT can perform the reverse mapping from ‘global’ to ‘local’ addresses.

Within a local network, IP addresses are generally assigned when a device is attached to the network, either manually or using a dynamic configuration protocol. Since addresses are topologically significant (for routing), a device’s address will change when it detaches and reattaches to the network elsewhere. Though such behaviour would not be expected in a single MANET, it would be likely to occur if a device was redeployed from one MANET to another.

The larger IPv6 address space opens up some new opportunities for different uses of the ‘local’ part of the address space. Typically, the 128 bit IPv6 address is split into a 64-bit ‘network identifier’ and a 64-bit ‘interface identifier’, the latter being the part that is unique per host on the local network. IPv6 neighbour discovery provides a mechanism for a host to select an address on the network, detect whether it conflicts with any existing devices and (depending on the address assignment mechanism) select an alternative address.

One use for the larger address space has been to place a cryptographic hash of a subject’s public key in the ‘interface identifier’ part of an address, to create a Cryptographically Generated Address (CGA) [5]. This is possible since the local network part of the address space is flat. An entity can assert ownership of an address by signing messages sent from this address using their private key.

In some cases IP addresses are used to identify a role rather than a single entity [9]. On a large scale, anycast can be used to provide robust infrastructure. The term ‘anycast’ refers to the routing of packets to the nearest instance of a particular IP address. Another example of non-unique ownership of IP addresses is provided by multicast addresses (and broadcast addresses in IPv4). For multicast, multiple hosts may join (or be implicitly part of) a multicast group. Traffic addressed to a multicast address is received by all of these hosts.

These issues concerning role-based addressing highlight that IP addresses are actually used for several different purposes. In fact, IP addresses actually play a triple role [12], [16]: an endpoint identifier, a routing object and a key value for forward lookup. This realisation has led to new developments in network protocol research over the last 20 years. This includes more recent work such as HIP [31]. In HIP, devices create a public/private key pair and create a long-term stable identity, called a Host Identity Tag (HIT), based on a fingerprint of the public key. This allows a node to assert ownership of an identity (HIT) as it moves around a network. The node still has a separate IP address, but this is used only as a locator, not an identifier.

Although it is desirable to attempt to distinguish the ‘identifier’ and ‘locator’ roles, the division is often blurred. One property we see as we look back down the layers is that a locator at one layer often becomes a (non-locator) identifier at the next lower layer. For example, an IP address is a locator at the network layer, but in the context of the link layer ARP protocol it simply becomes an identifier to lookup the link layer locator (MAC address).

Combined identifiers/locators at the network layer (such as IPv4 addresses) contain intrinsic weak protection against identity spoofing. Since the identifier/locator has topological/routing significance it is difficult for a remote attacker to spoof an identity in any useful way (the same is not true for a local attacker). If the identifier and locator are split, then this protection needs to be replaced with another mechanism to avoid opening up the possibility of easier identity spoofing. This is where the security mechanisms of protocols such as HIP are important to provide authentication of an identity assertion.

Since IP addresses may change and are not memorable, a more human-friendly name structure has evolved. DNS provides host-name-to-IP address mappings, creating an additional global namespace based on a hierarchical, federated structure. The standard DNS namespace has a single global root (‘.’). Names are constructed from sets of labels with dots between them. The namespace is delegated at zone boundaries (which are at the divisions between labels, though not all dots indicate a zone boundary). A complete lookup of a fully qualified domain name (FQDN) involves multiple requests, starting with a request to the root servers. Further queries are made to name servers where the namespace is delegated at zone boundaries.

C. Transport Layer

At the layers above IP the traditional transport protocols (such as TCP and UDP) contain further identifiers (port numbers) which, together with IP addresses and the protocol field in the IP header (the ‘5-tuple’) identify ‘flows’ between endpoints. Typically the destination port identifies a particular
service. The port (service) namespace is flat, although divided into sections based on standardisation requirements. There is general agreement on assignments in parts of the address space (well known and registered ports), while others (dynamic and private ports) are defined by local use.

D. Application Layer

At the application layer the common concepts are of ‘resources’ and ‘users’. The most common identifier construction is the Uniform Resource Identifier (URI). This is a multi-part identifier which provides information about protocol, user, server and resource. Examples for user contact include “mailto:someone@example.com” which identifies a user “someone” who can be contacted at host “example.com” using the service “mailto” (i.e. e-mail). The URI provides a globally unique namespace. The use of FQDNs as part of the resource identification means that it is intrinsically tied into the DNS infrastructure, and so builds on top of that hierarchical namespace.

E. What Does This All Mean For ID-PKC and MANETS?

The first conclusion to draw from this is that any useful IMS must be able to concurrently cope with both multiple namespaces and entities possessing multiple names within a namespace. Indeed, in some cases, a single name may also be owned by multiple entities. An IMS must also be able to manage dynamism in how identifiers are constructed. Some identifiers, such as MAC addresses, have a long term connection to an underlying device. Others, such as IP addresses are more likely to change over time as network structures adapt to new operational requirements. Although, from a network viewpoint, the ubiquity of IP addresses may suggest their use as a key form of identifier, their dual role as a locator may work against them, as IP addresses may change when a device’s point of attachment to the network changes. An IMS needs to be able to contend with this, by providing efficient identity repurposing and revocation mechanisms.

Where global hierarchical namespaces (e.g. DNS) form a key part of traditional networks, challenges arise in MANETs due to their dynamic nature, and possible disconnection from any infrastructure to provide the root of the hierarchy for name lookups. There may be a role for local or linked-local namespaces to provide resilient naming infrastructures, whilst hierarchical namespaces are still required for manageability.

Some work has been done on creating locator-independent identifiers based on public keys (HIP), and binding addresses to public keys (CGAs). Although some initial exploration has been done on using ID-PKC in this area [27], many of the real challenges have yet to be addressed. There is no standard structure for developing identifiers for ID-PKC for MANETs to incorporate the needs for identifying varying key lifetimes and managing revocation/re-purposing, structuring identifiers for hierarchical assignment, or managing the relationships between multiple namespaces.

V. ATTACKS ON IDENTITY INFRASTRUCTURES

Depending on the types of identity used at the different layers of a MANET, numerous attacks can be mounted against naming and addressing schemes. We detail two such attacks and briefly outline proposed defences to each. Where possible we indicate the constraints these defences place on the type of identifier available. Both of these attacks stem from the fact that identities can be created and discarded with equal ease. In this sense, many identities used in today’s networked environments follow the naming convention seen in Section II. Here identities are simply claims of ownership; it is establishing the veracity of this claim that causes difficulty and may allow an adversary to either impersonate another identity, generate multiple false identities all belonging to the same entity, or (due to the risk of capture of nodes in ad hoc networks) replicate the identity of captured nodes. For the interested reader, a more comprehensive study of attacks in MANETs can be found in [42]. In this section we are concerned with attacks related to abuse of identities.

Impersonation is when one entity falsely claims the identity of another entity. Impersonating the identity of another entity typically represents the first step for most attacks. These attacks are not only proactive in nature, in which an adversary initiates the impersonation of another entity, but can also be reactive.

Impersonation attacks are commonly seen in Man In The Middle (MITM) attacks on network protocols. In these attacks, an adversary intercepts communication between two peers. In doing so an adversary may alter the data sent from either sending or receiving peers without the knowledge of either. To defend against such attacks it is vital that both parties involved in a communication flow be authenticated. Authentication is typically based on some form of pre-establishment of authentication credentials through either knowledge of a shared secret or enrollment with a CA who issues credentials certifying an entity’s identity. An example of a CA issuing certificates to entities in MANET environments can be seen in theAuthenticated Routing for Ad hoc Networks (ARAN) protocol [36]. However, this approach provides no details as to who would act as a CA in a MANET environment nor does it provide details on how nodes enroll or are revoked.

A. The Sybil Attack

A Sybil attack is one in which a malicious party claims multiple identities, all of which are controlled by the same entity [19]. The ability to control an arbitrary fraction of the nodes in a network, allows a malicious adversary to effectively out-vote any honest nodes in collaborative tasks (such as Byzantine failure defences). It has been argued that in any P2P system without a centralised point of trust, such attacks on identity are endemic and can never be effectively combatted [19].

Recent approaches to combatting Sybil attacks have looked at using social networks [43] and trusted hardware [7]. In [7], the authors suggest that the use of trusted hardware can be used to constrain the number of identities an individual node
can obtain. In [43], trust relationships are established amongst user identities. The argument here is that a malicious user can create multiple identities but few human-established trust relationships. However, such an approach may be difficult to adopt in MANET environments as trust relationships established by autonomous technical nodes may dominate human-operated nodes in the network.

B. The Node Replication Attack

Node replication attacks are the dual of the Sybil attack. By capturing a node an adversary can replicate that node’s identity and distribute it throughout the network. The result of this attack is that several nodes will share the same identity which can result in routing protocols being led astray. In order to prevent such attacks Parno et al. [33] suggest the use of emergent properties of a network. They suggest two algorithms, randomized multicast and line-selected multicast to achieve this. In randomized multicast node location information is randomly distributed amongst randomly selected witnesses which exploit the birthday paradox to detect replicated nodes whilst line-selected multicast depends on topological information to detect multiple copies of the same node.

VI. Open Questions

Within a key management system, one of the most important features is the ability to bind identities to keys. However, in an ID-PKC setting the identities are the keys. This allows for a multitude of options in the type of identifier used. A number of difficult design options exist in deciding on suitable identifiers for an IKI scheme for MANETs. Traditionally, the choice of identifier has been application dependent, where an identifier would take the form of an email or IP address. However, as the research progressed and more complex information has been proposed for the inclusion in identifiers — information such as expiry dates [10], workflow information [14] or policy expressions [39] — it has become more difficult to interpret these identifiers. If there are no pre-arranged agreements as to the precise nature of what this extra identifying information means, how can one successfully interpret the intent of the owner of the identity?

Related to this are the joint issues of determining the correct identity for a given context, and the counterpart of determining the context to which an identity belongs. For example, an entity may take the role of a combatant in one operational scenario, and perform a humanitarian role in another. Although the identity of the entity is the same in both contexts, the context in which the identity operates define different rules of engagement. Issues also arise for identities within a protocol stack. As we said in Section IV-B, IP addresses act as both locators for points of network attachment at the network layer but also act as identifiers for endpoints at the transport layer. It is this repurposing of identity that makes the context in which an identity is interpreted of paramount importance.

As we saw in Section II, identities may change for no other reason than the personal preference of the identity owner. However, identities may also change when an entity migrates to a new area or when an entity joins a new group. In creating a new identity, it may be useful to establish the new identity by associating it with the old one. However, it may be important (in certain situations) that the privacy of the old identity be preserved. The disclosure of the linkage between the old identity and the new may be privileged information. Unlinkability is a property normally associated with anonymity systems, however, its relevance to a military setting may be strategic. A previous identity may be responsible for a particular function or be known capable of performing a particular action. It may not be desirable to disclose these linkages in all instances. In addition to this privacy concern, it may be a requirement that an identity should only be disclosed once certain conditions have been satisfied. In this regard, some identifiers should be “omni-directional” whilst others should be “unidirectional”. Omni-directional identifiers facilitate discovery, while unidirectional identifiers prevent the unnecessary release of correlation handles, here identity is disclosed at the owner’s discretion [11]. Related to this is the work of Back [6] on non-interactive forward secrecy and Anderson’s work on forward-secure signatures [3]. In [6], Back describes a system in which single (certified) public key is used to derive a sequence of public/private key pairs. The public keys form these pairs can be used as temporary identifiers for the user and can be deleted after a certain epoch.

In addition to changing identities in a single namespace over time, a device may have multiple active identities in different namespaces. These namespaces may be separate but related to one another as seen, for example, in the linkage between MAC addresses and IP addresses or HITs and IP addresses discussed in Section IV-B. The need for separate identities rather than identity reuse arises where identifiers are needed with different properties such as scope, structure, and rate of change.

There is also the open question of determining the correct identity of an entity one wishes to talk to. How does one go about discovering the identity of a remote peer in a distributed environment? To further complicate matters, once an identity has been discovered, there is the issue of identity ownership. ID-PKC assumes that we have implicit key authentication as one’s identity is one’s public key. Only the rightful owner of that identity should be able to decrypt a message sent to them. However, unless we have some means of authenticating an identity in the first place, any claimed identity by a peer is essentially meaningless.

Finally, there is the issue of revocation. Revocation in ID-PKC is typically handled by forced key refreshing. True revocation, however, in the case of compromise or malicious behaviour, is a much more difficult prospect. What does it mean to revoke someone’s identity? If, for example, our identifiers for our ID-PKC scheme were to be based on the IP address space [4], and one of the nodes in the network was detected as being malicious/compromised, what does it mean to revoke an IP address (particularly an IPv4 address)? Should that IP address never be reassigned to any other node via dynamic host configuration? How does the use of middleboxes
such as NATs or firewalls affect our revocation? As a corollary to this, we need to examine what happens to the old identity when it is abandoned in the case of IP address (identity) refreshing. All of these factors make the choice of identifier an oft overlooked, but crucial component in an IKI.

REFERENCES


