Child-proof Authentication for MIPv6 (CAM)

Greg O'Shea
Microsoft Research Ltd
St George House, 1 Guildhall Street
Cambridge CB2 3NH
+44 1223 744803
gregos@microsoft.com

Michael Roe
Microsoft Research Ltd
St George House, 1 Guildhall Street
Cambridge CB2 3NH
+44 1223 744744
mroe@microsoft.com

ABSTRACT

We present a unilateral authentication protocol for protecting IPv6 networks against abuse of mobile IPv6 primitives. A mobile node uses a partial hash of its public key for its IPv6 address. Our protocol integrates distribution of public keys and protects against falsification of network addresses. Our protocol is easy to implement, economic to deploy and lightweight in use. It is intended to enable experimentation with (mobile) IPv6 before the transition to a comprehensive IPSEC infrastructure.

General Terms

Security.

Keywords

Mobility, mobile communications, IPv6, IPNG.

1. INTRODUCTION

We describe a specialized security system for unilateral authentication of binding update messages in Mobile IPv6 (MIPv6) for use in the absence of a comprehensive IPSEC implementation [7, 8]. In this system a mobile node chooses a home address incorporating a cryptographic one-way hash of its public key, and ownership of a home address is established by demonstrating knowledge of the corresponding private key. Ownership is hard to falsify due to the difficulty of finding a key pair yielding a given hash, but is easy to verify and to enforce at critical points in the MIPv6 protocol. Replay detection is accomplished through loosely synchronized clocks. Above this we overlay an authentication and public key distribution protocol that is optimised for, and with minor modifications completely embedded within, the basic MIPv6 message exchange.

Protocols which construct addresses from a hash of a public key have been suggested in the past, for example by Phil Zimmerman, Carl Ellison, Christian Huitema and Jeff Schiller [6, p. 87]. Our protocol differs from previous proposals in that it does not use the entire hash, it integrates the distribution of public keys and it provides a form of access control to protect against falsification of network addresses rather than providing any generalized authentication service.

Our system builds upon various standard features found in IPv6 and IPSEC implementations, and is lightweight in the sense that it requires no manual configuration and incurs minimal message exchanges relative to IPSEC and IKE [2, 4].

We have observed an increasing interest in Mobile IPv6 and anticipate that deployment of Mobile IPv6 may outpace deployments of the IPSEC infrastructure that is expected to render it secure. This much is already apparent in some of today's experimental networks. For example, there is an administrative burden associated with IPSEC, requiring for example that security policy databases and certificates be installed on all participating hosts, and we are concerned that this overhead may impede the rate and extent to which MIPv6 is deployed. Equally we are concerned that implementations of MIPv6 that make no security provisions render entire IPv6 networks vulnerable to simple attacks. We therefore see a place for a basic but ubiquitous form of protection within mobile IPv6 to improve its security to the point where MIPv6, in the absence of IPSEC, can reasonably be regarded as no less secure than any other form of unauthenticated IP.

The remainder of this paper is organized as follows. In Section 2 we provide some essential background on Mobile IPv6 for the benefit of the reader who may not be familiar with this protocol. Section 3 presents the CAM protocol. In Section 4 we describe an ancillary security weakness occurring in the MIPv6 protocol, for which we propose a simple remedy based on access control. Section 5 suggests how the CAM protocol might be efficiently integrated into the MIPv6 message set. Our conclusions appear in Section 6.

The design of authentication protocols, even simple ones, is notoriously error prone. We have therefore undertaken a manual analysis of CAM using the logic of Burrows, Abadi and Needham [1]. We present the results in an appendix.

2. BACKGROUND

In conventional circumstances the IPv6 address of a host encodes its whereabouts in terms of a distinct network link so that packets can be routed in its direction. In contrast, MIPv6 describes mechanisms whereby a *mobile node* can freely roam between network links and yet constantly remain accessible through a *home address* statically allocated on its "home" network. While away from its home network a mobile node makes use of a *care-of address* dynamically allocated on the network to which it is currently attached, while a proxy known as a *home agent* is responsible for forwarding (tunnelling) packets arriving at the home network on to the mobile node's care-of address.

Conference '00, Month 1-2, 2000, City, State.

A mobile node makes its whereabouts known to its home agent by sending it a *binding update* message giving its home address, its current care-of address and the lifetime for which the binding should be honoured. Every home agent maintains a *binding cache* recording the binding updates it has received. Any node that communicates with a mobile node is a *correspondent node*, and in fact every IPv6 node is a potential correspondent node. A mobile node also sends a binding update to any correspondent node from which a (tunnelled) packet has been received via its home agent. Each correspondent node (optionally) maintains a binding cache which its transmit function uses to redirect packets directly to the mobile node's care-of address, thus saving at least one network hop relative to the route through the home agent. Home agents and correspondent nodes may refresh a binding cache entry by sending a binding request to a mobile node, requesting the transmission of a fresh binding update.

MIPv6 minimises the state that a correspondent node must maintain by including a *home address option* field, encoding a mobile node's home address, in every packet sent by a mobile node while it is away from its home network. The receive side of the IP stack on the correspondent node informs higher level software that the packet was sourced not from the care-of address in the packet's header, but from the address given in the home address option.

In the absence of a reliable authentication mechanism it is extremely easy to fabricate bogus binding updates and home address options; the effort involved can amount to little more than manually assigning the target IPv6 address to an MIPv6 node while disconnected from the network, then attaching the node to any network that will accept its connection.

MIPv6 mandates the use of IPSEC authentication (IPSEC AH) for binding updates and binding acknowledgements (replies thereto) [9]. In particular this prevents an impostor from causing all traffic destined for a mobile node to be misdirected (for example, to the impostor) by submitting bogus binding updates. The motivation for authentication of binding acknowledgements is to defend against denial of service attacks, which we consider in Section 3. The MIPv6 draft allows all other message types to go unauthenticated, including those containing a home address option.

In passing we observe that Mobile IPv4 describes authenticators formed from a hash of message fields and a shared secret. Our approach differs fundamentally in that we do not require shared secrets.

The goal of CAM is to provide a minimum level of unilateral authentication of binding updates when IPSEC AH is not available. This is reflected in its name. We strongly recommend the use of IPSEC AH for protecting binding updates wherever practicable; indeed there is a strong case for employing IPSEC AH on all traffic between a mobile node and a correspondent node whenever a full IPSEC security association exists between them. However, the use of IPSEC AH on binding updates (and binding acknowledgements) is sufficient to satisfy the current draft and we are not alone in anticipating that some hosts may wish to limit themselves to precisely this.

A minimal realistic exchange between two MIPv6 nodes may comprise one message pair, for example an ICMP6 (echo, reply) pair. CAM is relatively efficient in such cases.

3. THE CAM PROTOCOL

When mobile CAM node is first initialised it creates a (public, private) key pair, which it immediately saves it to secure local storage. It next chooses a candidate home address for itself (strictly, an aggregatable global unicast address). The routable (high order) 64-bit address prefix is obtained by listening for local router advertisements, in the normal manner [5, 12]. It is common practice, albeit unenforced, to derive the remaining (low order) 64-bit *Interface Id* from the interface's MAC (EUI-64 global identifier) address. Interface Ids such as these are economic to generate and have the attractive property of often being globally unique. In CAM, however, the Interface Id is a cryptographic one-way hash of the node's public key (in fact, the leading bits generated by the SHA1 algorithm). The "u" and "g" bits of an IPv6 Interface Id have the semantics defined for EUI-64 global identifiers, and are therefore always zeros in CAM so that a CAM Interface Id is not mistaken for a globally unique EUI-64 identifier [5, p8]. The hash of the public key distributes evenly across the remaining 62 bits of the Interface Id.

Our approach requires the use of a hash algorithm that is one-way (non-invertible) and the selection of large enough part of the hash to render inversion of the hash infeasible. Note that it is reasonable to use only part of the hash because an opponent must compromise a given hash (of a node's public key); in comparison modern hash functions are designed to resist the much easier attack in which it suffices to compromise an arbitrary hash. We cannot protect against a user who deliberately generates two keys that hash to the same address, but we are not seeking to provide non-repudiation in this protocol.

We believe that the use of a 62-bit value is adequate for most low-end requirements, comparing favourably with the 40-bit keys commonly used for SSL in web-based applications, particularly as it is harder to generate and test a key pair than it is to test an SSL session key. In addition, although a given key must be retained for the duration of any existing TCP connections, it should be practical to change keys at intervals of several days. Existing DNS servers should cope with the resultant updates. During the transition to a new key, the previous key (and associated Home Address) can remain in use.

It is possible, albeit improbable, for the hash of a public key to coincide with an existing Interface Id sharing the same link, for example a manually assigned address or the Interface Id of another CAM node whose public key (which may or may not be different) has the same hash. IPv6 *duplicate address detection* is used to detect this [12]. Moreover, a mobile node may wish to use its public key to form a new home address on some arbitrary link at some point in the future, so we provide a means whereby address conflicts can be resolved without requiring the mobile node to either change its public key or to hold multiple public keys. The mobile node generates and remembers a *modifier, i*, for each link on which it holds a home address. The modifier is appended to the public key prior to forming the hash, and if an address conflict occurs then a different modifier is used and the operation is repeated until a link-unique Interface Id is obtained. It is important to keep the size of the modifier as small as possible. One or two bits is sufficient to make the probability of an unresolvable collision extremely low; a larger modifier would weaken the protocol by making it easier to find a colliding hash because the attacker can try each key several times.

At this point everything is set for the mobile node's binding updates to be authenticated by any CAM-enabled correspondent node. Observe that no manual configuration is involved whatsoever. This makes deployment easy, and compares favourably with the administrative overheads of configuring IPSEC databases or deploying certificates.

A run of the CAM protocol is initiated by a mobile node that wishes to send a binding update to a correspondent node (which may be a home agent) with which it cannot authenticate using IPSEC. Note that this inability to authenticate using IPSEC AH does not prevent either node from employing an IPSEC security policy database dictating what interaction, if any, is permitted to occur between them.

A complete run of the CAM protocol requires just one message, causing minimal disruption to application flows. The correspondent sets its own policy towards the permitted difference in timestamps. We silently drop rejected messages to prevent their replay by an impostor.

We now describe the CAM protocol using the following notation: M and C are principals (mobile and correspondent, respectively), A'_m is M's care-of address, A_c is C's address, (PK_m, SK_m) is M's (public, private) key pair, i is the modifier used to resolve name clashes, H(m) is a hash of m, T_m is M's time-stamp, $\{m\}SK_m$ is a signature of m using key SK_m, R is the route prefix of M's home address and A_m = R, $H(PK_m, i)$ is M's home address.

$$M \rightarrow C : A'_m, A_c, A_m, PK_m, i, T_m, \{H(A'_m, A_c, A_m, T_m)\}SK_m$$

This message incorporates M's public key, which C may not know in advance, together with the modifier, i. Initially C compares T_m against its own clock to confirm timeliness, and may reject the message at this point. C verifies that $A_m = H(PK_m, i)$ in other words that PK_m is a public key associated with that address (recall that several such keys may exist). C next uses PK_m to recover $H(A'_m, A_c, A_m, T_m)$, generates its own hash $H(A'_m, A_c, A_m, T_m)$ and rejects the message if the two hashes differ. C now believes that the sender knew the secret (key) associated with the home address, and accepts this as proof of message authenticity.

Observe that A'_m, A_c and A_m are all bound by $H(A'_m, A_c, A_m, T_m)$, so that old messages cannot be used in a replay attack involving some other combination of addresses.

Our use of uncertified public keys may appear unusual, but has some precedent [6, p87]. In this case it is reasonable because the key pair is not associated with any real-world identity; it is merely associated with an IPv6 address and is used solely as a way of making it expensive for an impostor to forge a binding update for a given Home Address without prior knowledge of the appropriate secret key. An impostor wishing to compromise C would need to discover an alternate key pair hashing to M's home address, which is prevented by properties of the cryptographic hash, or it would need to discover SK_m, or it would need to conduct a replay attack within a small amount of time allowed by C.

Applications above the IP layer may wish to make use of the results of the CAM protocol, for example using addresses that have been verified using CAM within access control lists or when establishing network connections; how applications might do this is outside the scope of this paper. In particular, CAM does not assure the trustworthiness of (e.g. DNS) names or name to address bindings; applications that depend upon such things should employ other mechanisms to establish their reliability.

We do not defend against denial of service attacks in which the attacker attempts to saturate the recipient by sending a large number of Binding Updates; if this is a concern then alternative protocols such as IKE should be deployed. The following denial of service attack, which was suggested by one of the anonymous referees, warrants some consideration: the attacker causes binding updates to be dropped in the network (for example by jamming the network) and returns unauthenticated Binding Acknowledgements to the mobile node. This may result in the home agent and correspondent nodes holding stale entries in their Binding Caches and sending packets to an old foreign network, where the packets will be lost. Where this attack is of concern we recommend that the home agent and mobile node establish and use an IPSec security association to protect (at least) all Binding Updates and Binding Acknowledgements. The number of Security Associations involved is sufficiently small to make key distribution practical. Another possibility is that correspondent nodes, and home agents, may protect their Binding Acknowledgements using a straightforward variant of CAM.

At one stage we considered a variant of this protocol that did not require loosely synchronised clocks. We abandoned this approach as it resulted in the need to hold state or extended protocol runs. The use of clocks permits a much simpler solution.

4. HOME ADDRESS OPTION

An impostor may send a packet to C including a home address option citing A_m as the source of the packet. In the absence of authentication any such packet will appear to have originated at M, potentially damaging C, or M, or both. Furthermore, if M has sent binding updates to C (and/or to M's Home Agent) and if the impostor and M are transmitting on the same port (by accident or design) then C may send a reply to the impostor's message directly to M, perhaps circumventing firewalls that prevent the impostor from attacking M directly.

This attack is easy to demonstrate using current (experimental) MIPv6 software. Doing so requires little more than a configuration utility to reset the impostor's IPv6 home address.

We propose that every MIPv6 correspondent node C defend against this attack by dropping every unauthenticated packet that contains a home address option citing home address A_m sent from address A'_m for which no entry (A_m, A'_m) exists in its binding cache. On receipt of any such packet C should issue a binding request to M or send a packet to M via M's home agent, to ensure that the binding cache at C is up to date for M. If the dropped packet was in fact sent by M then the ensuing binding update creates an entry (A_m, A'_m) in C's binding cache and the packet will be accepted if it is retransmitted (automatic for TCP). On the other hand if the dropped packet was not from M then a probable attack has been thwarted.

We observe that this defence is easily defeated by an impostor who can submit packets to a correspondent node using a source address equal to the care-of address of a legitimate mobile node. This, however, is the general weakness of unauthenticated IP and is not due to Mobile IPv6.

5. INTEGRATING CAM INTO MIPV6

In this section we suggest how CAM may integrate into the MIPv6 message set. This turns out to be straightforward, thanks mainly to the foresight shown by the (mobile) IPv6 protocol designers in designing extensible message formats.

We adhere to the IPv6 convention that the order of elements within a packet reflects the order within which they should be processed upon receipt. In particular, any packet containing a binding update also contains a home address option, but not vice versa, and we expect that the home address option will precede the binding update in any such packet.

Consider the very first packet sent from a mobile node, M, to a correspondent, C. The packet contains a home address option and a binding update, but C lacks the requisite binding cache entry. We propose that a new IPv6 *destination sub-option type* be defined for use within the home address option, announcing the presence of a binding update deeper within the packet. The effect is to delay the packet drop that would otherwise occur immediately for any unauthenticated home address option (see Section 4).

The elements A'_m, A_c and A_m are IPv6 addresses each of which occupies 128 bits. T_m is the high order 32 bits of SNTP encoding (i.e. seconds) [10]. The modifier *i* is encoded in 16 bits. PK_m and $\{H(A'_m, A_c, A_m, T_m)\}SK_m$ both have length 128 bytes, including padding. We propose that these be carried, in the order shown, within the binding update option as new destination sub-option types. We observe that the sub-option length field permits destination sub-options of up to 255 bytes in length, which is ample for our purposes.

Given existing code for parsing IPv6 message elements, together with the cryptographic routines widely available in support of IPSEC, we do not anticipate that CAM should prove difficult to implement.

To keep the implementation lightweight and to facilitate interoperability we prescribe the use of 1024-bit RSA for signing and SHA-1 for the public key hash. This avoids the overheads of a framework within which the use of alternative algorithms may be negotiated, and should not leave an uncomfortable legacy if IPSEC proves successful.

6. CONCLUSIONS

We have described an authentication protocol designed specifically to expedite experimentation with Mobile IPv6 by reducing the risks of deploying MIPv6 without full support for IPSEC AH.

CAM is designed so that it is easy to implement, economic to deploy and lightweight in use. In the optimum case it makes extensive use of existing IPSEC routines, requires no manual administration and generates no additional network packets.

We intend to undertake an experimental implementation of CAM based upon the MSR IPv6 stack and the MIPv6 extensions thereto made at Lancaster University (http://www.LandMARC.net/).

We would stress that CAM considers only the authenticity of binding updates and home address options. No assurances are provided concerning other information sent from a mobile to a correspondent, and no assurances are made concerning information sent from a correspondent to a mobile at all. In circumstances in which only a small number of packets require authentication, the increased packet size used in CAM compares well with the overheads of an IKE protocol run. In circumstances where a large number of packets require authentication, the use of IKE and IPSEC AH will be more efficient.

The principles employed in CAM can also be used to solve other security problems. For example, we are currently investigating the use of a CAM-like protocol as the basis for establishing IPSEC security associations.

7. ACKNOWLEDGMENTS

Our thanks to Andrew Scott, Stefan Schmid, Joe Finney, Doug Shepherd, Richard Black, Christian Huitema, Roger Needham, Richard Draves, Art Shelest, Dieter Gollmann, Cedric Fournet and the anonymous referees for their comments and suggestions on a draft of this note.

8. REFERENCES

[1] Burrows, M., Abadi, M. & Needham, R. M. 1989 A Logic of Authentication. Proceedings of the Royal Society of London Series A, 426, pp. 233 – 271.

[2] S. Deering, S. and Hinden, R. Internet Protocol, Version 6 (IPv6) Specification. Internet Engineering Task Force (IETF) RFC2460. December 1998.

[3] Federal Information Processing Standard FIPS PUB 180-1, Secure Hash Standard. NIST. April 1995. http://csrc.nist.goc/fips/fip180-1.txt

[4] Harkins, D. and Carrel, D. The Internet Key Exchange (IKE). Internet Engineering Task Force (IETF) RFC2409. November 1998.

[5] Hinden, R. and S. Deering, S. IP Version 6 Addressing Architecture. Internet Engineering Task Force (IETF) RFC2372. July 1988.

[6] Huitema, C. IPv6 The New Internet Protocol, Prentice Hall PTR, ISBN 0-13-850505-5, 1998.

[7] Johnson, D. and Perkins, C. Mobility Support in IPv6. Internet Engineering Task Force (IETF) draft-ietf-mobileip-ipv6-12.txt. 27 April 2000.

[8] Kent, S. and R. Atkinson, R. Security Architecture for the Internet Protocol. Internet Engineering Task Force (IETF) RFC2401. November 1998.

[9] Kent, S. and Atkinson, R. IP Authentication Header. Internet Engineering Task Force (IETF) RFC2402. November 1998.

[10] Mills, D. Simple Network Time Protocol (SNMP) Version 4 for IPv4, IPv6 and OSI. Internet Engineering Task Force (IETF) RFC2030. October 1996.

[11] Perkins, C. IP Mobility Support. Internet Engineering Task Force (IETF) RFC2002. October 1996.

[12] Thomson, S. and Narten, T. IPv6 Stateless Address Autoconfiguration. Internet Engineering Task Force (IETF) RFC2462. December 1998.

9. APPENDIX : BAN ANALYSIS OF CAM

We present the results from a manual analysis of CAM using the BAN logic [1]. We make use of the notation and the informal description of CAM from Section 3 above, and include references to BAN derivation rules.

A principal in the BAN logic starts with initial belief in at least one key or secret. Such initial beliefs are expressed as assumptions, and the BAN logic does not describe how these initial beliefs are established. From this standpoint there is nothing unusual about our assumption $C \models PK_m$ below. From another standpoint the assumption warrants some justification because C starts with no knowledge of PK_m. We believe that the assumption is justifiable because we are using addresses derived algorithmically from keys, which avoids the need to maintain an address to public key binding.

The goal of the protocol is:

$C \models M \models A'_m$

In other words that C believes M believes A'_m to be M's care-of address.

We make the following assumptions:

$C \models \#(T_m)$ [A1: C believes M's timestamp is fresh]

$C \models PK_m \mapsto M$ [A2: C believes M has public key PM_m]

Strictly speaking, $C \models PK_m \mapsto M$ results from C regenerating the hash of PK_m and M's demonstration that it knows SK_m. We cannot infer this in BAN, so we provide assumption A2 instead.

Starting with the message described in Section 3, we proceed as follows:

$\mathbf{M \rightarrow C : A'_m, A_c, A_m, PK_m, \mathit{i}, T_m, \{H(A'_m, A_c, A_m, T_m)\}SK_m}$

$C \triangleleft H(A'_m, A_c, A_m, T_m)SK_m$ [1: immediate]

$C \models M \mid\sim A'_m, A_c, A_m, T_m$ [2:from 1, A2, message-meaning rule]

$C \models \#(A'_m, A_c, A_m, T_m)$ [3: from A1 and part-fresh rule]

$C \models M \models A'_m, A_c, A_m, T_m$ [4: from 2, 3, and nonce-verify rule]

$C \models M \models A'_m$