KONGSBERG

CoNSIS

# MIKE assessment

# Technical Report

Document ID: 1/1559/1-FCPR10127 Rev B

Authors:
Anne Marie Hegland
Hans-Are Ellingsrud

# Summary

Basically, the Multicast Internet Key Exchange MIKE does for multicast groups what the Internet Key Exchange (IKEv2) does for IPsec unicast peers. It authenticates new group members and automates the establishment of a cryptographic symmetric group key.

The report assesses MIKE according to the criteria; Security, Availability, Bandwidth efficiency, Robustness. Other criteria such as maturity of the protocol and documentation, scope of use, necessary preconditions for the protocol to work and power efficiency, are also discussed. The report focuses on the use of MIKE in tactical ad hoc networks and identifies a number of possible improvements.

Two modes of operation are specified for MIKE: In the Key Agreement mode all group members contribute to the group key through multiple Diffie-Hellman iterations. Once the group key has been established, one node is appointed transaction manager. The transaction manager is responsible for new inclusions and exclusion of members. The other mode – the Key Distribution mode – is a centralized scheme where a key distribution centre generates and distributes the keys. This resembles the traditional symmetric key management scheme of pre-placed keys and a central key distribution centre, but provides a more efficient distribution of new keys by organizing the keys in a key tree. In addition MIKE includes a standard public key based three-way mutual authentication of new group members.

The assessment indicates that the Join and Leave protocols are vulnerable to replay attacks. A stronger linking between the messages in the Join protocol, and better replay protection of the Leave protocol, is needed for MIKE to prove secure under a formal security analysis.

A major challenge for both modes of operation is the dependency upon reliable multicast. Group members that loose an update packet are in effect expelled. Forward Error Correction and timers that trigger re-join on missed key updates have been implemented to counter this. Periodic repetition of the latest update, as this report suggests, will also increase the robustness. But none of the measures fully solve the problem. Other proposed enhancements include allowing keys to overlap in time and change of group keys only on eject in Key Distribution mode.

The Key Agreement mode has been proposed for tactical networks, and the Key Distribution mode for strategic networks. A conclusion of the report is that the Key Distribution mode is the more suitable also for tactical ad hoc networks. The compulsory key change when the group changes, accidental exclusions by lost update packets and the disruption caused by the re-join delay represent a threat to network availability in Key Agreement in particular.

MIKE consumes no bandwidth in a stable group. It is the group changes that cause the traffic. Multi-hop multicasts are very bandwidth consuming. Bandwidth can be saved by multicasting only the strictly necessary keys. Additional suggestions for bandwidth savings include: Skip the LeaveConfirm message, collapse the TMdistribute and UpdateDistribute messages in Key Agreement mode. The distribution of certificate revocation information only to the key distribution centre in Key Distribution mode is also suggested to save bandwidth.

Another finding is that MIKE cannot easily be used as the only key management scheme in a tactical multi-hop ad hoc network. This is a tactical communication nodes will normally not forward traffic from non-group members, i.e. those nodes that do not have the group key. Unless a group key has been pre-distributed, or the key distribution centre / transaction manager is within the 1-hop neighbourhood of the joining node, the newcomer will not be able to affiliate to the network.

The documentation of MIKE is still immature. This was a challenge in the assessment. We thank the inventor, Dr. Aurisch at the Fraunhofer institutes, for valuable clarifications.

Our conclusion is that MIKE is not clearly superior to a traditional pre-placed symmetric group key scheme. The major benefit is its ability to include new members ad hoc.

# Table of Contents

## Figures



## Tables

# 1. Introduction

Much of the communication at the lower tactical echelons is multicast radio traffic by nature. Position data for friendly force tracking must be distributed to all coalition partners within shooting range. Friendly forces that move into the area need the proper group key to start receiving position data. Other examples are sensor data and alarms. The tactical army group need a common cryptographic key to protect their radio communications. This calls for an efficient group key management scheme. The Multicast Internet Key Exchange (MIKE) is one candidate scheme.

Basically MIKE does for multicast groups what the Internet Key Exchange (IKEv2) [11] does for unicast peers: It provides automated negotiation of group security parameters such as keys and cryptographic algorithms for IPsec. But MIKE has not reached the same level of maturity and standardization. In this paper we study its applicability for the lower tactical echelons and discuss some possible enhancements.

## 1.1 Scope

The aim of this assessment is to answer the question to what extent is MIKE applicable in tactical mobile ad hoc networks, and to reveal benefits as well as unsolved challenges and possible enhancements.

MIKE is often used in combination with an IPsec discovery protocol (IDP) to automate IPsec multicast. The IDP discovers IPsec peers, and MIKE establishes a common group key. However, the focal point of this report is MIKE. IDP is not studied here.

## 1.2 Identification

This document describes the work and results from the MIKE assessment provided by Kongsberg Defence & Aerospace (KDA) under Norwegian national funding within the Coalition Network for Secure Information Sharing (CoNSIS) project and under contract between Forsvarets Forskningsinstitutt (FFI) and KDA [1]. FFI coordinates the Norwegian participation in the CoNSIS project. A number of private subcontractors have been engaged to provide parts of the work. KDA is one of the subcontractors.

The target of the MIKE assessment activity is an assessment of the MIKE key management protocol for use in tactical ad hoc networks. The report is based on a theoretical study of available literature and discussions with the designers of the scheme.

## 1.3 Related work

The efficiency of MIKE is analysed in [8] focusing on message sizes and delay. The article presents categories of requirements with resemblance to the evaluation criteria used in our assessment, but they only to some extent overlap. Reference [8] discusses security (forward and backward secrecy and access control), efficiency (scalability and multiple requests), dependability (fault tolerance and robust key update) and other criteria such as performance under EMCON and real-time characteristics. This assessment puts stronger emphasis on tactical ad hoc networks and evaluates MIKE according to additional criteria. Some requirements in [8], such as backward secrecy, are considered less important in our scenario. The work brings in additional perspectives to those presented in the analysis in [8]. In addition, a number of improvements are suggested.

## 1.4 Abbreviations and definitions

| | |
|---|---|
| AES | Advanced Encryption Standard |
| AH | Authentication Header |
| AHA | All-Hear-All (1-hop network) |
| CA | Certificate Authority |
| CRL | Certificate Revocation List |
| COI | Communities Of Interest |
| CoNSIS | Coalition Networks for Secure Information Sharing |
| DH | Diffie-Hellman |
| DoS | Denial of Service |
| EMCON | Emission Control |
| FFI | Forsvarets Forskningsinstitutt |
| ID | IDentity |
| IDP | IPsec Discovery Protocol |
| IEEE | Institute of Electrical and Electronic Engineers |
| IKE | Internet Key Exchange |
| IKEv2 | Internet Key Exchange version 2 |
| IP | Internet Protocol |
| IPsec | IP Security Protocol |
| IV | Initialisation Vector |
| KDA | Kongsberg Defence & Aerospace |
| KDC | Key Distribution Centre |
| MAC | Media Access Control |
| MANET | Mobile Ad-hock Network |
| MIKE | Multicast Internet Key Exchange |
| MOU | Memorandum of Understanding |
| MPR | Multi-point Relay |
| OLSR | Optimized Link State Routing protocol |
| OSPF | Open Shortest Path First |
| PKI | Public Key Infrastructure |
| PPK | Pre-Placed Key |
| SA | Security Association |
| TM | Transaction Manager |
| UHF | Ultra High Frequency |
| VHF | Very High Frequency |

## 1.5 Document overview

We start with an introduction to MIKE. Then our scenario is described. Then the assessment criteria and MIKE assessment are presented. Finally our suggestions for improvements and concluding remarks are included.

## 1.6 References and related documents

[1] Technical Arrangement - TA Number 2009.01 between FFI and KDA Concerning KDA deliverables to the CoNSIS project, Annex B to Collaboration Agreement

[2] CoNSIS MOU, 03-06-09

[3] T. Clausen (Ed.), and P. Jacquet (Ed.), "Optimized Link State Routing Protocol (OLSR)," RFC 3626, 2003.

[4] R. Ogier, and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding," RFC 5614, 2009.

[5] R. Coltun, D. Ferguson, J. Moy and A. Lindem (Ed.), "OSPF for IPv6," RFC5340,2008

[6] T. Aurisch, "Using Key Trees for Securing Military Multicast Communication," IEEE MILCOM 2004.

[7] T. Aurisch, "Optimization Techniques for Military Multicast Key Management," IEEE MILCOM 2005.

[8] T. Aurisch, T. Ginzler, and P. Martini, "Practical Efficiency Analysis of a Dual Mode Group Key Management", IEEE MILCOM 2008.

[9] T. Aurisch, T. Ginzler, and P. Martini, "Practical Efficiency Analysis of a Tree-based Dual Mode Group Key Management," ACM 1-58113-000-0, unpublished manuscript, 2007

[10] T. Aurisch, T. Ginzler, P. Martini. R. Ogden, T. Tran, and H. Seifert,"Automatic Multicast IPsec by using a Proactive IPsec Discovery Protocol and a Group Key Management," Journal of Telecommunications and Information Technology, No.2, 2008.

[11] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)," IETF RFC 5996, 2010.

[12] K. P. Kihlstrom, L. E. Moser, P. M. Melliar-Smith, "The SecureRing Protocols for Securing Group Communication," In Proceedings of the 31th Annual Hawaii International Conference on System Sciences, 1998, pp. 317-326.

[13] T. Aurisch, "Tree-based Dual Mode Group Key Management for Improving Key Establishment Scalability," FKIE, unpublished manuscript.

[14] T. Aurisch, and J. Krajewski, "System Specification of the IPD MIKE System", Version 0.2 - English translation, work in progress, 2010.

[15] Y. Xiao, and J. Rosdahl, "Throughput and Delay Limits of IEEE 802.11," IEEE Communications letters, Vol.6, No. 8, August 2002.

[16] E. Winjum, A. M. Hegland, P. Spilling, and Ø. Kure, "A Performance Evaluation of Security Schemes proposed for the OLSR Protocol," MILCOM, 2005.

# 2. Introduction to MIKE

MIKE exploits key trees for efficient management of keys. MIKE can operate in either Key Agreement or Key Distribution mode of operation. The main difference between the two modes of operation is the way that auxiliary keys are generated. In the Key Distribution mode of operation, a Key Distribution Centre (KDC) is responsible for the management of the key tree and the generation and distribution of keys in the tree. In the Key Agreement mode, all users contribute in the generation of the group key and all must have a common view of the key tree. Reference [7] suggests Key Agreement mode of operation for tactical use and Key Distribution mode for strategic networks. Both modes of operation demand a Public Key Infrastructure (PKI).

## 2.1 Key trees

In Key Distribution mode the key tree is a logical construct known only by the KDC. In Key Agreement mode of operation, the users maintain a common perception of the key tree.

The following gives a brief overview of the use of key trees with focus on the Key Distribution mode.

Efficient key distribution is a main benefit of key trees. Instead of encrypting the new group key with the individual keys of all N group members it is encrypted with auxiliary keys shared by subsets of the users.

The key tree consists of two types of nodes: key nodes representing keys and user leaves representing users. Figure 1 illustrates the concept. A user (N) knows only the keys on the direct path from his leaf to the root node (highlighted). The root node is the group key. User leaves contain unique individual keys. The other nodes contain auxiliary keys.

Figure 1 illustrates that all users/group members (N1-N8) possess the group key $K_{12345678}$. The auxiliary (subgroup) key $K_{1234}$ is common to N1- N4, and $K_{12}$ is common to N1 and N2. $K_1 – K_8$ refer to the individual users' keys. Assuming user N8 is to be revoked; all group and auxiliary keys known to N8 ($K_{12345678,}$ $K_{5678}$ and $K_{78}$) must be updated. N7 shares all auxiliary keys with N8. N7 therefore receives all updated keys encrypted with its individual key. The new group key and auxiliary key can be distributed to N5 and N6 encrypted with their common auxiliary key; $K_{56}$. To N1-N4, the group manager sends the new group key $K_{1234567}$ encrypted with auxiliary key $K_{1234}$. Thus, bandwidth and computational cost is saved compared to traditional distribution where the new group key is encrypted with the individual keys of each remaining group member.

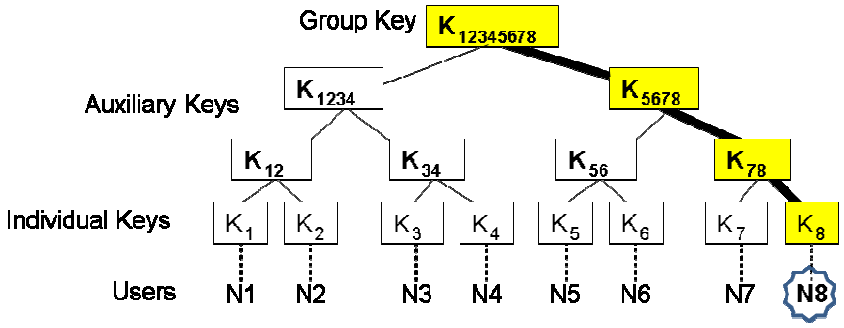



**Figure 1 Key tree**

The key tree can be binary, as shown in the figure, or k-ary, and be balanced or unbalanced.

## 2.2 Key Distribution mode of operation

This mode requires a central KDC. The KDC itself is normally not a group member. The users have no knowledge about the key tree. The KDC uses the tree to organize the keys. The users receive the common group key and their needed auxiliary keys from the KDC.

The authenticity of the messages in the protocol between the users and the KDC is assured through digital signatures. The KDC signs the distribution message with its private key. A necessary precondition is that all members have an authenticated version of the KDC's public key, or the certificate must be included in the distribution message. In either case some pre-configuration is necessary: the public key of the KDC or the root certificate must be distributed in advance. The basic operations are user Join and user Leave.

MIKE resembles the well known approach of symmetrical key distribution using pre-placed keys (PPK) and a Key Distribution Centre (KDC) with the difference that MIKE offers better efficiency due to the use of the key tree construct and exploits asymmetrical methods for authentication of new users and integrity protection of the protocol.

### 2.2.1 Key Distribution mode, Join protocol

Figure 2 outlines the Key Distribution MIKE Join protocol. It consists of five steps. New users receive the group key from the KDC after completing an authenticated Diffie-Hellman key agreement during a standard three-way mutual authentication procedure. Then the KDC sends the group key and necessary auxiliary keys to the newcomer in step 4. In step 5 the group key is updated and multicast to all the group members. A necessary prerequisite is static identification numbers for each user, e.g. their global IPv6 addresses.

Step 1: The user initiates the protocol by unicasting a signed "JoinRequest" message to the KDC. The join request message includes information about the user identity, the group security association for which the group key is need, and information about cryptographic algorithms supported. The latter two are called SA in the figure. The signed JoinRequest message also includes the user's public Diffie-Hellman value and a nonce for replay detection.

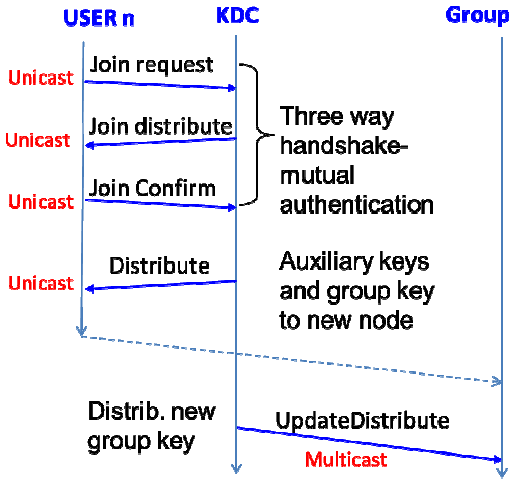
Step 2: The KDC returns its own public Diffie-Hellman value in a signed "JoinDistribute" message. The message also includes the identity of the KDC and a new nonce, plus information received from the user in the join message. That is, the nonce, the users' public Diffie-Hellman value and SA information.

Step 3: The Diffie-Hellman based shared secret key has now been established between the joining user and the KDC. The user completes the three-way mutually authenticated Diffie-Hellman key agreement by returning a unicast "JoinConfirm" message. It consists of a signature calculated over the nonce issued by the user and the nonce received from the KDC and his public Diffie-Hellman value plus the group security association and supported cryptographic algorithms.

Step 4: The KDC unicast the group key and necessary auxiliary keys to the joining user through the "Distribute" message. The information is encrypted with the shared secret from step 3, and the message is signed with the aid of the private key of the KDC. The group key can be a new one or the current one – depending on the group policy.

Step 5: If the group policy demands backward secrecy, the last step is key update. The KDC then renews the group key and relevant auxiliary keys, and multicasts it to all group members in the "UpdateDistribute" message. The updated keys are encrypted with the auxiliary keys

where possible, and encrypted with individual user keys to those nodes close to the new user in the key tree. The signed UpdateDistribute message includes the identity of the KDC, group security association and cryptographic algorithm information and sequence number – all encrypted with the group key, and the new key table and the signature of the complete message.




| Step | | Message | |
|---|---|---|---|
| 1 | Join Request | $M1,\sigma\{M1\}_{priv_n}$ | $M1=\{ID_n,\ SA,\ \eta_n,\ DH_n\ \}$ |
| 2 | Join Distribute | $M2,\sigma\{M2,\ \eta_{n1}\}_{priv_{KDC}}$ | $M2=\{ID_{KDC},\ SA,\ \eta_{KDC},\ DH_{KDC}\}$ |
| 3 | Join Confirm | $\sigma\{M3\}_{priv_n}$ | $M3=\{ID_n,\ SA,\ \eta_n,\ \eta_{KDC},\ DH_{KDC}\}$ |
| 4 | Distribute | $E(M4)_{k_{DH}},\sigma\{M4\}_{priv_{KDC}}$ | $M4=\{ID_{KDC},\ SA,\ Seq,\ KT'\}$ |
| 5 | UpdateDistribute | $E(M5)_{k_{group}},\ KT'',\ \sigma\{M5,\ KT''\}_{priv_{KDC}}$ | $M5=\{ID_{KDC},\ SA,\ Seq+1\ \}$ |

**Notation:**

n = user n
$ID_x$ = identity of node x
Seq = Sequence number
SA = security association information
η = nonce
$\sigma\{x\}_k$ = Message x signed with key k
$E(x)_k$ = Message x encrypted with key k

$priv_n$ = Private key of user n
$priv_{KDC}$ = Private key of KDC
$DH_n$ = Public Diffie-Hellman value of user n : $g^a \bmod p$
$DH_{KDC}$ = Public Diffie-Hellman value of KDC: $g^b \bmod p$
$k_{DH}$ = shared secret Diffie-Hellman key $g^{ab} \bmod p$
$k_{group}$ = (old) group key
KT' = Group key and auxiliary keys needed by new user n
KT" = New group key and new auxiliary keys needed by group

**Figure 2 Outline of MIKE Join protocol for Key Distribution mode of operation**

### 2.2.2 Key Distribution mode, Leave protocol

Figure 3 outlines the Leave protocol. The exiting user initiates a leave by unicasting a leave request to the KDC. The KDC confirms the leave request and multicasts new keys to the remaining members of the group. That is, the KDC can choose to rekey immediately, or postpone the update until more leave requests have been received. Such batched membership operations can optimize the distribution when bursts of requests can be expected.

The KDC can also eject a node independently of leave requests. This is done by distributing a new group key and new auxiliary keys on the path from the user to be ejected to the root in an UpdateDistribute message.

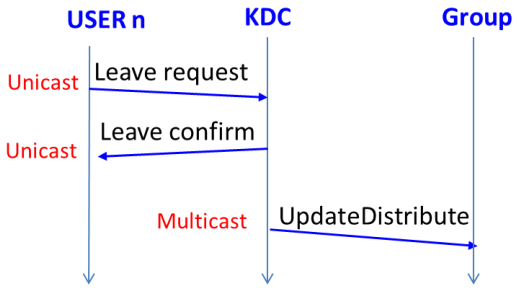



**Figure 3 Outline of MIKE Leave protocol for Key Distribution mode of operation**

Whereas the parameters of the Key Distribution mode Join protocol are specified in [6], there are fewer details on Leave in available literature. We assume that the UpdateDistribute message of the Leave protocol is identical to the UpdateDistribute message of the Join protocol shown in Figure 2.

## 2.3 Key Agreement Mode of operation

In this mode all users participate in the generation of the group key. Differently from the Key Distribution Mode of operation, all users must know and agree on the key tree. The key tree is built through iterative Diffie-Hellman key agreements. Users N1 and N2 establish a common secret key $K_{12}= g^{ab} mod\ p$ where *a* and *b* are secrets known only by N1 and N2, respectively. The generator *g* and the prime modulus *p* are known in advance. In the same way users N3 and N4 establish a common secret key, $K_{34}= g^{cd} mod\ p$. The *blinded* version of this key, $g^{K_{34}} mod\ p$, is exchanged with users N1 and N2 and vice versa. N1 and N2 as well as N3 and N4 can then calculate the next key on the path to the root, $K_{1234}= g^{K_{12} K_{34}} mod\ p$. But N1 and N2 can not reveal $K_{34}$, likewise N3 and N4 cannot reveal $K_{12}$. Through these iterative Diffie-Hellman exchanges, all members learn the group key and key tree, but no party get to know all keys in the tree.

One node is appointed as Transaction Manager (TM). The TM is responsible for processing subsequent Join and Leave requests.

Figure 4 shows the join and leave protocols for the MIKE Key Agreement mode of operation after the initial group key establishment. To Join, the new user sends its public Diffie-Hellman value to the TM in a tree-way mutual authentication procedure identical to the one used between the user and the KDC in the Key Distribution mode. The newcomer does not know who the current TM is, but posts the JoinRequest to a predefined multicast address. Only the TM answers this request. After the three-way handshake, the new node is included in the tree, and the TM transfers the TM role to the newcomer in the "TMDistribute message". In same message, it also redistributes the key tree with all unaltered blind keys. The new TM calculates the tree path by using Diffie-Hellman multiple times. Then the new TM multicast the new blind keys on the path from itself to the root in the "UpdateDistribute" message. Every user can now calculate the group key.

Note that various articles on MIKE present slightly different protocols for key agreement. Some include an extra Distribute message before TMDistribute or a TM confirmation message after it. Figure 4 is based on the description in [14]. For additional information on the variants see APPENDIX D.

A Leave is initiated by the exiting user. The user sends a Leave request to the TM. The TM confirms the Leave and removes the user from the key tree. It also sends an UpdateDistribute message that enables the remaining users to calculate the new group key.

The TM has the power to eject nodes without Leave request. If the TM leaves, it must transfer the TM role to one of the remaining group-members before it exits.

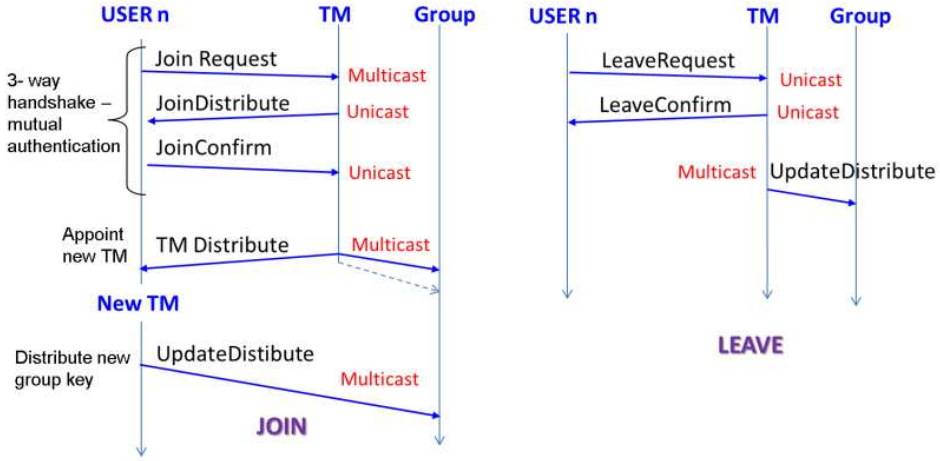



**Figure 4 Join and Leave protocols in MIKE Key Agreement mode of operation**

Missing an UpdateDistribute message has the same effect as being ejected. On Joins, all group members therefore start a timer when a JoinRequest multicast is heard. If the UpdateDistribute has not been received within the expected time frame, the group member assumes it has lost the new group key and initiates a new JoinRequest. To what extent the LeaveRequest is also sent as a multicast that enables the members to start their timer, or unicast (as illustrated in the figure) is not clearly stated in the specifications. The description in [14] is interpreted as a unicast.

The procedure "AgreeOnTM" is initiated in case the TM is lost and a new TM must be elected. The AgreeOnTM is based on SecureRing communication [12]. First the users flood presence information. All users send a Join message. After some Join messages all users get a common view of the available members. Then they send Commit messages which trigger the election of a new TM. A token is circled, and the TM is uniquely defined during the process. A more detailed explanation of AgreeOnTM was not found.

## 2.4 Changing the mode of operation

According to [8] MIKE can dynamically change from Key Agreement to Key Distribution mode in order to adapt to the environment when efficiency problems due to increasing group size or decreased transmission capacity occur.

In Key Agreement mode, no member – including the TM - knows all keys in the key tree. A major difference between the TM and the KDC role is that the KDC knows all keys in the key tree.

As the KDC learns all keys, it cannot normally be part of the key tree. Consequently, when going from Key Agreement mode to Key Distribution mode a new entity must take over as KDC, or the group member that takes over as KDC must be expelled from the group.

The system specification in [14] states that the user that takes the role as KDC expels itself from the key tree and establishes a new key tree using repeated DH iterations. It is not explained exactly how the parties agree on the change of operation and how the KDC should be elected. The current implementation requires manual configuration.

The KDC needs to establish pair-wise unique keys with each member of the group. This necessitates a Diffie-Hellman key agreement with each of the group members. Then the necessary auxiliary keys can be distributed.

The KDC should establish the key tree as a logic construct and distribute the necessary auxiliary symmetric keys to the users. The users need not know the key tree. However, [14] assumes that the KDC continue to use the iterated DH key tree as in the key agreement mode. That is, the KDC distributes blind keys that enable the group members to calculate the auxiliary keys (resembling a TM that continues to act as a TM after it has left the group). This demands that the users continue to know at least of “their” part of the key tree: they must be aware of the path to the root.

The change from Key Distribution to Key Agreement mode is only briefly outlined in the literature.

## 3. Our scenario: mobile ad hoc network for the lower tactical echelons

Figure 5 illustrates our scenario. We assume a multi-hop mobile ad hoc network (MANET) for the lower tactical echelons. The network consists of heterogeneous VHF or UHF wireless tactical communication nodes. The nodes[1] are at the same time both routers and end-hosts. Some are vehicle mounted. Others are battery powered and carried by dismounted soldiers. The nodes differ in level of mobility as well as in power resources and transmission range. Some of the nodes may also need to enter radio silence (EMCON mode) for a shorter or longer period. There may or may not be a connection to deployable infrastructure. The focus is on the MANET.

Communication is protected by a group key. New members of the network must have the proper group key in order to communicate. The group key can be pre-placed. But there is also a need for including new members ad hoc. One example is when coalition partners come within shooting range. Then they should start receiving position data for friendly force tracking. The number of nodes in the wireless network is typically from 10 to 50.

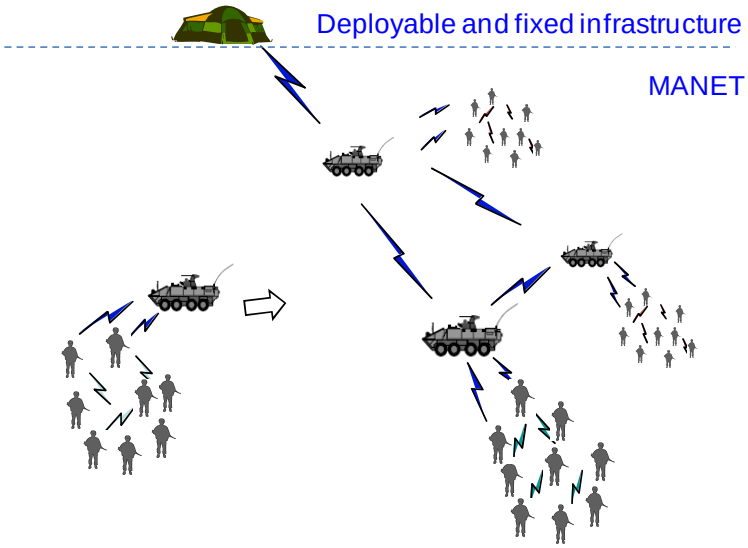



Figure 5 Scenario: Wireless communication at the lower tactical echelons

[1] The term “node” here refers to a wireless communication node in the tactical ad hoc network, and will be used interchangeably with the terms “user” and “group member”. Note the distinction from nodes in the key tree. There are more nodes than users in the key tree. The context will show whether we refer to a node=user in the wireless tactical network or a node/user in the key tree

# 4. Assessment criteria

## 4.1 Security

**Secure Protocol:** The protocol itself must be secure. It must withstand attacks on the protocol messages and the order of messages. It must not fail if put under analyse using a formal verification method. The protocol should also be robust to insiders that do not behave according to the protocol.

Proper cryptographic primitives are a necessary precondition. But as the cryptographic primitives can easily be replaced by others with the proper strength, emphasis is put on the protocol rather than the primitives in the sequel.

**Forward secrecy:** Forward secrecy is important in the sense that it must be possible to expel compromised nodes and leave them incapable of learning new keys through their knowledge of earlier keys. Backward secrecy when new members join is less important. Hiding earlier information from a friendly, authorized user when he joins the network is rarely needed. Besides, much of the information exchanged at the lower tactical levels has only short-lived value.

## 4.2 Availability

Whereas availability is reckoned an integral part of security, it is so important in the tactical scenario that it is here treated separately.

**Add new members dynamically:** Pre-configuration is to a large extent possible in the tactical scenario, but it must also be possible to add newcomers and friendly forces ad hoc.

**Seamless addition of new member and key changes:** It must be possible to add new group members and change keys without the users experiencing disruption. It should be possible to include new members without having to change the group key. Nodes that are prevented from taking actively part in transmissions due to radio silence must not be excluded due to a group key change.

**Delay:** Group membership changes and key updates must complete in a timely manner. The delay must not be longer than that the previous key change/update completes before the next one starts.

## 4.3 Bandwidth efficiency

A natural concern in a wireless environment is the channel occupation. The protocol must scale to the expected group size. As a rule of thumb, management traffic should not occupy more than 10% of the available bandwidth. Being only a fraction of the management traffic, MIKE should not use more than 1-2% of the available bandwidth.

## 4.4 Robustness

**Robust to link losses:** Varying connectivity and temporary outages can be expected. The protocol must be robust to spurious link losses. It must survive Denial of Service (DoS) and replay attacks without disrupting the communication.

**No single point of failure:** The network must be operable even if it is partitioned or specific nodes are temporarily unreachable.

## 4.5 Other

Other assessment criteria include parameters such as **maturity** of the protocol and documentation, intended (and possible) **scope of use,** necessary **preconditions** for the protocol to work and **power efficiency**.

# 5. MIKE Assessment

Table 1 summarizes the assessment of MIKE described in the next sections. The table indicates to what extent we found that the specific requirement is fulfilled.

The most obvious alternative of MIKE is the well known approach of pre-placed keys (PPK). That is, pair-wise symmetrical keys for protection of the communication between each member and the KDC, as well as a common group key. (The initial pre-placed group key is used until the first member is expelled.). The PPK w/KDC is included in the rightmost column of the table for comparison.

**Table 1 Result of the Assessment**

| Criteria | Key Distribution | Key Agreement | PPK w/KDC |
|---|---|---|---|
| Secure Protocol | Partially | Partially | Yes |
| Forward Secrecy | Yes | Yes | Yes |
| Add members dynamically | Yes | Yes | No |
| Seamless key change | No | No | Yes |
| Seamless add new member | Partially | No | Yes |
| BW efficient | Partially | No | Partially |
| Robust to link loss | Partially | No | Yes |
| No single point of failure | No | Yes | No |
| Power efficient | Yes | No | Yes |
| Mature | No | No | Yes |
| Scope of Use | Small to large network<br>Separation of COI<br>Unsuitable as initial key | Small network<br>Separation of COI<br>Unsuitable as initial key | Small to large network<br>Separation of COI<br>OK as initial key |
| Preconditions | Trust relation exists<br>PKI<br>Running network service | Trust relation exists<br>PKI<br>Running network service | Pre-distribution |

## 5.1 Security

**Secure protocol:** The formal security analysis of MIKE using a formal verification tool such as AVISPA or similar, is outside the scope and time limit of this assessment, but some considerations are:

*Link the three-way handshake to the rest of the Join protocol:* A nonce or sequence number that links the first three steps of the protocol to the fourth should be added. The Seq in step 4 in Figure 2 serves as a freshness "anchor" that later receptions can be compared to. The sequence number (Seq) in steps 4 and 5 enables the user to determine the freshness of messages from the KDC. However, from the description in [6] it is not obvious how the joining user can verify the freshness of the message received in step 4. That is, the message is encrypted with the unique key established in steps 1-3, and the user knows what plaintext some of the fields should contain. But without a nonce or sequence number that links the first three steps with the fourth step, we suspect that the MIKE join protocol may fail to prove secure under a formal verification. This applies both for the Key Distribution and the Key Agreement modes of operation.

*Leave's replay vulnerability:* The Leave request message is prone to replay attacks. The KDC/TM cannot easily determine the freshness of a Leave Request.

*Vulnerability against insiders/ DoS attacks:* From the available documentation it is not evident to what extent MIKE protects against insiders that by accident or deliberately do not behave according to protocol. Apparently, Key Agreement is more vulnerable to such behaviour than the Key Distribution mode. One or more nodes that repeatedly try to join and rejoin may impose a constantly changing key. The change of TM when a new member joins also makes it easier for the illicit node to take over control. We assume that the protocol prevents multiple nodes from posing at TM at the same time as the others expect the previous one to appoint the new.

Altogether we find that the requirements for a secure protocol are only partially fulfilled, as illustrated in Table 1.

The MIKE documentation provides few details on the cryptographic primitives and key lengths. Apparently X.509 certificates (or similar) are used, and the shared secret established with the TM or KDC is used as a group key for some symmetric algorithm.

**Forward secrecy:** MIKE enables both forward and backward secrecy. In Key Distribution mode of operation the KDC decides whether the keys are updated or not on a change in the group. This is beneficial. In the wireless environment it is hard to guarantee that all nodes receive the new key in a timely manner. Unnecessary key changes should be avoided. They cost bandwidth and represent a threat to availability - valid members that missed the key update have to re-join in order to continue their communication.

In Key Agreement mode both forward and backward secrecy come intrinsically – it is not possible to include or exclude a node without changing the group key. Basically, bandwidth and availability is traded for backward secrecy. This is undesirable in the tactical scenario. The Key Agreement mode of operation should be used with great caution in such networks. Key Distribution mode of operation is generally considered a better option where possible. MIKE meets the forward secrecy requirement as shown in Table 1.

## 5.2 Availability

**Dynamically addition of new members:** Both modes of operation enable dynamic inclusion of new group members, but only if they already have an established trust relationship. The new member and the TM/KDC must present a certificate that the other party accepts. The certificates must be signed by a certificate authority that the other party knows and trusts.

**Seamless key changes and seamless addition of new members:** The Key Distribution mode of operation enables joins without rekeying. The Key Agreement mode does not. Both protocols are vulnerable to packet losses. Nodes in radio silence that miss the transmission are cut off until they exit the radio silence mode and again are able to re-join.

On this background we find that MIKE allows dynamic addition of new members, but does fully meet the requirements for seamless key – and group changes, as pointed out in Table 1.

**Delay:** Delays from the initiation of a Join or Leave operation until all nodes have received the new group key include processing delays due to cryptographic operations as well as transmission delays. The total end-to-end delays are studied in [6], [8] and [13]. The repeated DH operations in the Key Agreement mode make it significantly more resource consuming than Key Distribution mode with increasing number of group members (users/leaf nodes). The auxiliary keys used in the Key Distribution mode are also shorter than DH values. Consequently the Key Agreement mode gives longer delays than the Key Distribution mode.

The join delay is very important as MIKE requires users that missed a key update must re-join to continue their communication. Long disruptions are not acceptable. In VHF networks, the transmission delay represents a larger fraction of the total delay than in UHF networks. In UHF networks the processing delay is more important. The processing delays can be reduced by adding more processing power. Little can be done to reduce the transmission delays accordingly.

## 5.3 Bandwidth efficiency

Once the group key has been established, MIKE consumes little or no bandwidth. It is the group changes caused by Joins and Leaves that contribute to the bandwidth consumption. In our scenario, we expect that the number of group changes during an operation will be limited. But it is hard to decide the frequency of Joins and Leaves. We here take a different approach: we estimate how often a Join or Leave operation can take place without exhausting the channel. That is, we calculate the time the channel is occupied by a Join operation, i.e. from the new node sends its Join request message to the end of the UpdateDistribute message.

Differently from [6], [8] and [13] that provide calculations and simulation results for MIKE focusing on total delay including processing delay, the focus is the transmission delay – or rather – the time the channel is occupied by transmissions of MIKE protocol messages. We also include the effect of overhead added by the lower layers of the protocol stack and certificate distribution. Furthermore, we study the consequence of multi-hop communication.

The channel occupation is calculated for different numbers of nodes in the network, considering both a 1Mbps UHF and 20kbps VHF network. The calculations encounter both a 1-hop all-hear-all network and multi-hop networks. And we compare the Key Distribution and Key Agreement modes of operation. The resulting channel occupation refers to the bandwidth consumed within the 1-hop neighbourhood of the transmitting node. (This is an optimistic scenario.)

The calculations assume MIKE messages are encapsulated in UDP over IPv6. IEEE802.11b is the MAC protocol used for UHF calculations. For VHF a proprietary MAC protocol has been used. The IEEE 802.11b adds a delay of 552µs per MAC frame [15], whereas the proprietary VHF protocol is assumed to add a delay of 125ms per MAC frame. For simplicity, the UHF calculations assume that all messages – not only multicast messages – are sent with the IEEE802.11b broadcast data rate of 1Mbps. To reduce the bit error rate at the IP-layer to an acceptable level a Forward Error Control (FEC) system is assumed at the link layer. This FEC adds 20% overhead to all MIKE messages.

The assumed message formats used in the calculations are described in APPENDIX A. Constants are documented in APPENDIX C. The calculations assume an error free channel at the IP layer due to the FEC and no collisions.

The estimates for multi-hop networks assume that multicast messages are forwarded by multi-point relay nodes (MPRs) that have been appointed by a routing protocol such as OLSR [3] or other protocol. MPR nodes are chosen so that when these nodes forward the traffic, 2-hop neighbours of the transmitting node is covered. More details on MRPs are provided in [3]. The numbers of MPRs for various network sizes used in our estimates is based on the network topology and simulations results in [16].

The multi-hop calculations are based on the assumption that the KDC/TM is located within 1-hop range of the joining node. That is, unicasts to the KDC/TM need not be re-transmitted (forwarded). Multicast messages, on the other hand, are forwarded by the MPRs.

Figure 6 shows the results. Figure 6a) compares UHF to VHF 1-hop networks including both Key Distribution and Key Agreement modes of operation. The figure demonstrates that a Join operation in Key Agreement occupies the channel significantly longer than a Join in Key Distribution mode.

Figure 6b) highlights the effect multi-hop networks. The figure includes both Key Distribution and Key Agreement modes of operation with 16 to 48 nodes as this is what we have multi-hop simulation data for. The transmission times of a Join operation increases significantly when going from a UHF 1-hop to a multi-hop networks. It also shows that a Join in Key Distribution mode in a multi-hop network demands approximately the same amount of bandwidth as a Key Agreement Join in a 1-hop network.

The simulation in [6] indicates that Key Distribution mode of operation performs well in networks with the characteristics of Ethernet and ISDN, but has problems in lower bandwidth networks such as VHF. Our calculations support this. The time the channel is occupied is significantly longer in VHF networks. The figures also show that Key agreement in UHF multi-hop networks introduces transmission delays (and thus re-join delays for nodes that were ejected by accident due to packet loss) in an order of magnitude that can cause unacceptably long disruption of the communication.

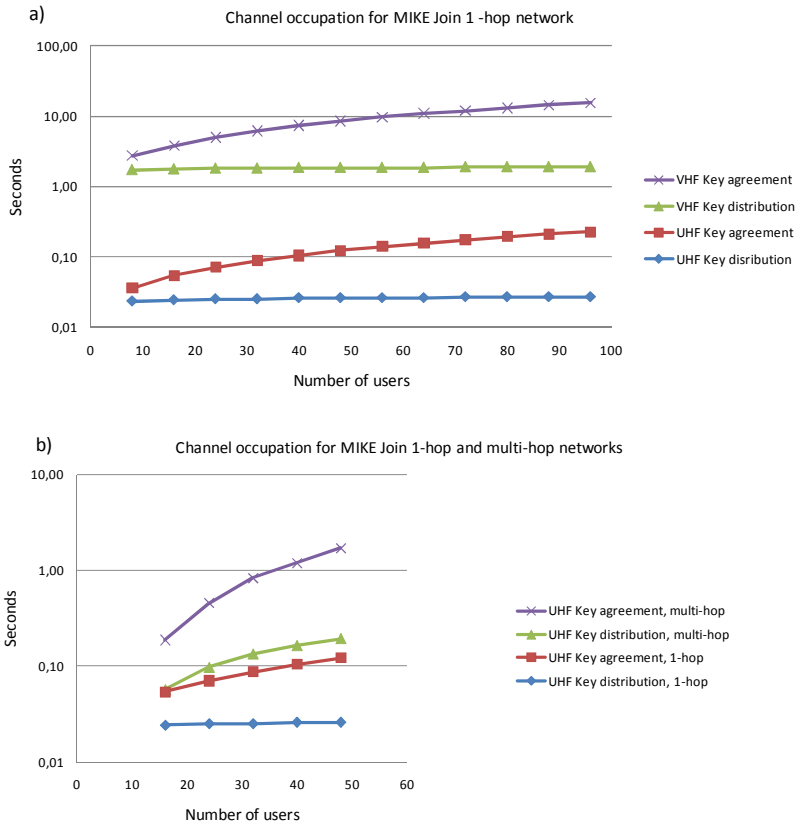



**Figure 6 MIKE Join Transmission times under the assumption of no other traffic**

## 5.4 Robustness

MIKE demands reliable multicast. This is hard to achieve. A single packet loss can cause unintended ejection of group members. Key Agreement cannot include new users without changing the key. Key Distribution can be made more robust by allowing group changes without key changes. The timers, acknowledged unicasts and forward error correction mechanisms that have been included in MIKE only to some extent remedy the problems caused by packet losses and bit errors.

### Protection against link losses:

*Timeouts:* If the expected response message is not heard within the timeout interval after a join request, the node tries to re-join. Whereas this mechanism may be acceptable in those cases where a single node suffered from temporary loss of network connectivity, it also has the potential to trigger a multicast storm. If the JoinRequest was heard, but the TM suffered from network connectivity problems, a major part of the nodes in the network may try to re-join at the same time. This can lead to congestion and cause additional problems.

Furthermore, whereas JoinRequest are multicast messages and hence enable the group members to start their timers waiting for the UpdateDistribute, LeaveRequests are unicast. LeaveRequests also

lead to key changes. But the group members cannot calculate when to expect such an UpdateDistribute message as the LeaveRequests are not heard by others than the nodes on the route between the leaving node and the KDC/TM.

*FEC:* The correct delivery of multicast messages such as the Update/UpdateDistribute messages is not guaranteed. The FECs increase the probability of successful reception at the IP layer despite bit errors at the physical layer for radios close to the reception threshold. The drawbacks are that FEC adds overhead, and it does not help if the packet collided or was lost due hidden or exposed node problems. Nodes in radio silence must wait until they exit this mode before they can re-join.

**TM vulnerability:** The Key Agreement approach of appointing the joining node as the next TM represents a threat to availability and robustness. It favours communication nodes with the lowest network connectivity as TMs. Communication nodes at the edges of the radio net are more prone to lost packets than the well connected nodes. Consequently the probability is higher that they need to re-join and thus become TM.

Furthermore, the TM must perform two cryptographic operations for each node on the root path. It is better that this is done by more powerful vehicle nodes rather than resource constrained solider nodes. In addition vehicle mounted nodes usually have higher output power and thus a longer transmission range. Another problem is that newcomers are also more likely to exit the network again shortly[2]. The node that has been TM for a long period will probably remain in the network, and should continue to act as TM.

Transferral of the TM role to the latest joining or re-joining node is not an appropriate approach in tactical ad hoc networks.

## 5.5 Other

**Maturity - documentation:**

MIKE is documented through a number of publications [6][7][8][10][13][14]. The protocols vary slightly from publication to publication. See APPENDIX D for further details. The assessment assumes that the latest [14] applies. At least one implementation of MIKE exists. None of the publications include enough details to implement compatible versions.

Contradicting specifications needs to be resolved. The description in [14] indicates that the users calculate the tree also in the Key Distribution mode. This is unnecessary when the tree is a logic construct only needed by the KDC. However, the users must be able to distinguish between auxiliary keys and the group key. Auxiliary keys are only used for the communication with the KDC. The group key shall be used for encryption and decryption of user data between the nodes in the wireless network. A key reference with some indication of key type is needed.

**Applicability - scope of use:**

***MIKE as the initial key management scheme:*** MIKE requires an already running network service or a one-hop network. A newcomer will otherwise not be able to communicate with the KDC or the TM. The nodes in the tactical ad hoc network are at the same time both routers and end hosts. The nodes

[2] Thanks to Pierre Simon at cogisys for pointing this out

should not forward traffic from unauthenticated nodes, and new nodes are not included unless the link has been authenticated. Tactical ad hoc networks typically rely on link-encryption for the protection of the wireless links. IPsec could be used instead. But a pre-shared key scheme will in either case likely be needed in addition to MIKE in order to protect the wireless link and to provide a secure "bootstrap" of the network service.

If we could assume that any joining node will always be within direct transmission range of the KDC/TM, MIKE could be used to establish this basic group key used to protect the wireless links. But this assumption does not apply to multi-hop networks in general. Consequently, MIKE cannot easily be used as the only key management scheme in tactical ad hoc networks. It is more suitable for establishing group keys for separation of communities of interest (COI)/secure multicast groups "on top of" the already protected links.

***Mode of operation in tactical ad hoc networks:*** Whereas [7] suggest Key Agreement mode of operation for tactical use and Key Distribution for strategic networks, we believe that the Key Distribution mode will perform better also for tactical use. Both modes of operation depend on the availability of a central entity – the KDC or the TM. The Key Agreement mode demands more bandwidth. It is not possible to include a new node without re-keying. All nodes must maintain a common view of the key tree, and the operations such as agreeing on a new TM demands multiple rounds of transmissions from the members.

Key Agreement can be used as a fall back in the situation where the army group is isolated from deployable and fixed infrastructure or access to the KDC for other reasons is not possible.

**Preconditions:**

***PKI:*** The security of MIKE rests on the public keys exchanged in the three way handshake. MIKE demands a Public Key Infrastructure and a pre-shared root certificate. Alternatively the public keys of the participants can be pre-distributed. In the Key Agreement mode, all members need to know the public keys of the others. In Key Distribution mode, it suffices that any joining member knows the public key of the KDC. The KDC needs to know the public key of any user that may join the group.

***A running network service*** is a necessary precondition for joining users' communication with the KDC/TM.

***Power efficiency*****:** The change of TM in the Key Agreement mode for every join means that every newcomer must be prepared to take this role independently of whether this is a battery powered soldier node or less resource constrained vehicle mounted node. This is both undesirable and unnecessary. Only well connected and the least resource constrained nodes should be given this role.

## 5.6 A note on MIKE compared to a pre-placed key

Table 1 compares MIKE with a traditional pre-placed key (PPK) scheme.

***Security:*** The PPK scheme is secure under the assumption that the group key and pair-wise symmetric keys were transferred over a secure channel – possibly distributed manually with a courier.

***Forward secrecy:*** is obtained by distributing a new group key encrypted with the pair-wise unique keys.

***Dynamic addition of new members:*** It is not easy to add new members dynamically as the key must be pre-distributed.

***Seamless key changes*:** Once the group key and pair-wise keys are pre-distributed, it is possible to change the keys while retaining the previous ones until all has received the new.

***Seamless additions of new members*:** It is possible to distribute the group key to a new node without demanding that the others change their key(s).

***Bandwidth efficiency*:** Pair-wise symmetric keys are not known to scale well when the group becomes large. Though, the symmetric keys are small compared to certificates and signatures. For the expected group size in of our scenario, PPK is assumed to fulfil the bandwidth efficiency requirement partially.

***Robust to link losses*:** The PPK scheme can be made robust to link-losses by repeating the latest key update periodically or by the use of positive acknowledgements.

***No single point of failure*:** A PPK scheme with a central entity (KDC), does not meet this requirement.

***Power efficiency*:** Schemes that rely solely on symmetric cryptographic operations are less resource consuming than public key schemes.

***Maturity*:** PPK is a mature and well tested scheme.

***Scope of use*:** A pre-placed key is well suited as the initial key. More keys can be used in order to separate communities of interest.

Altogether, the traditional PPK w/KDC scheme fulfils most of the requirements, but lacks MIKE's possibility of including new members dynamically. In addition, it does not include MIKE Key Agreement’s robustness against single points of failure.

# 6. Proposed optimizations of MIKE

## 6.1 New optimizations of MIKE

Table 2 summarizes optimizations that were identified during the assessment and outlines their implications. As shown in the table, some relate to both modes of operation, others to only one of the operation modes. The "+" sign indicate what aspects the suggestion improves. A "-" indicate that the suggestion has a negative consequence on this criterion.

**Table 2 Outline of possible optimizations of MIKE and their impact**

| Criteria / Suggestions | Security | | Availability | | Robustness | | Other |
|---|---|---|---|---|---|---|---|
| | Secure Protocol | Forward secrecy | Seam-less key change | BW efficiency | Robust to link loss | No single point of failure | Power efficiency |
| **General** | | | | | | | |
| Enhance replay protection | + | | | | + | | |
| Retransmit last key | | | + | -/+ | + | | |
| Allow key overlap | | - | + | | + | | |
| Skip LeaveConfirm | | | | + | | | |
| Introduce backup KDC/TM | | | | | | + | |
| **Key Distribution mode** | | | | | | | |
| Distribute CRL only to KDC | | | | + | | | |
| Change group key only on ejects | | + | | + | + | | |
| **Key Agreement mode** | | | | | | | |
| Collapse TMDistribute and UpdateDistribute | | | | + | | | |
| Add TM willingness | | | | + | + | | + |
| Do not transmit entire key tree on Join or Leave | | | | + | | | |

### 6.1.1 Enhance replay protection

We suggest that the KDC and TM uses sequence numbers instead of an arbitrary nonce in the three-way handshake, and in addition includes the updated sequence number in steps 4 and 5 of the Join protocol. This links the three-way handshake to the next messages of the Join protocol. The sequence number enables each group member to detect whether a message from the KDC/TM is fresh or not. The group members always keep a copy of the newest sequence number received in the latest multicast. In Key Agreement mode, when a new node takes over the TM role, it continues to increment the sequence number last received from the previous TM. (The joining node can continue to use random nonces or its own sequence number.)

In order to reduce the vulnerability to replay attacks, we also suggest that a timestamp is included in the LeaveRequest message or that the Leave is authenticated trough a three-way handshake.

### 6.1.2 Retransmit last Key

Nodes who discovers they have lost the group key, must re-join. Robustness may be improved by periodically repeating the last UpdateDistribute message. As long as only one node missed the UpdateDistribute a re-join may be the more efficient approach. But in the case that more nodes missed it, repeating it can improve robustness. The rate of the repetitions must be weighed against the added bandwidth cost. Repetitions will likely be most important just after the key change. After each repetition, more nodes will probably have received the update.

The repeated transmissions consume extra bandwidth. But the method also reduces the probability of frequent re-joins. This pulls in the other direction when calculating the resulting total bandwidth consumption.

We have calculated the channel occupation caused by repeating the UpdateDistribute messages periodically at different intervals with varying numbers of nodes in the network, assuming that the FEC is able to correct all errors at the IP layer and no collisions.

Figure 7 shows the average channel occupation for different types of radio nets under varying conditions. Figure 7a), b) and c) show a 1-hop UHF net. Figure 7d), e) and f) shows a multi-hop UHF net, and Figure 7g), h) and i) a 1-hop VHF net. Figure 7a), d) and g) show the average channel occupation for Key Agreement when the whole key tree is distributed. Figure 7b), e) and h) show the average channel occupation for Key Agreement when only the modified blind keys are distributed. Figure 7c), f) and i) show the average channel occupation for Key Distribution.

The figures show that under the assumption that MIKE should not take up more than 1-2% of the total channel capacity, the acceptable repetition frequency depend on network type as well as group size and mode of operation.

Table 3 summarizes the results. It indicates that periodically repetitions is a viable approach in UHF networks – especially for Key distribution mode and for Key agreement when only the updated blinded keys are repeated. In VHF networks it should be used with great caution.

**Table 3 Estimated minimum time between re-distributions in order not to exceed 1-2% of the bandwidth**

| | | Operation mode | | |
|---|---|---|---|---|
| | | KAM | Opt KAM | KDC |
| UHF | 1-hop | 20s | 2s | 1s |
| | Multi-hop | 120s | 20s | 10s |
| VHF | 1-hop | 1200s | 120s | 120s |
| | Multi-hop | - | - | - |

KAM = Key agreement - re-distribute whole key tree
Opt KAM = Key agreement - re-distribute only updated blind keys
KDC = Key distribution - re-distribute group key and new aux keys

We have assumed an ideal situation where the FEC is able to correct all errors at the IP-level. In a realistic scenario errors will still occur after FEC decoding. When the IP-packet is reassembled and contains errors it will be dropped. A retransmission of the same packet may contain another error and the packet may be dropped again. Assembling correct FEC-blocks from different transmission can diminish this problem.

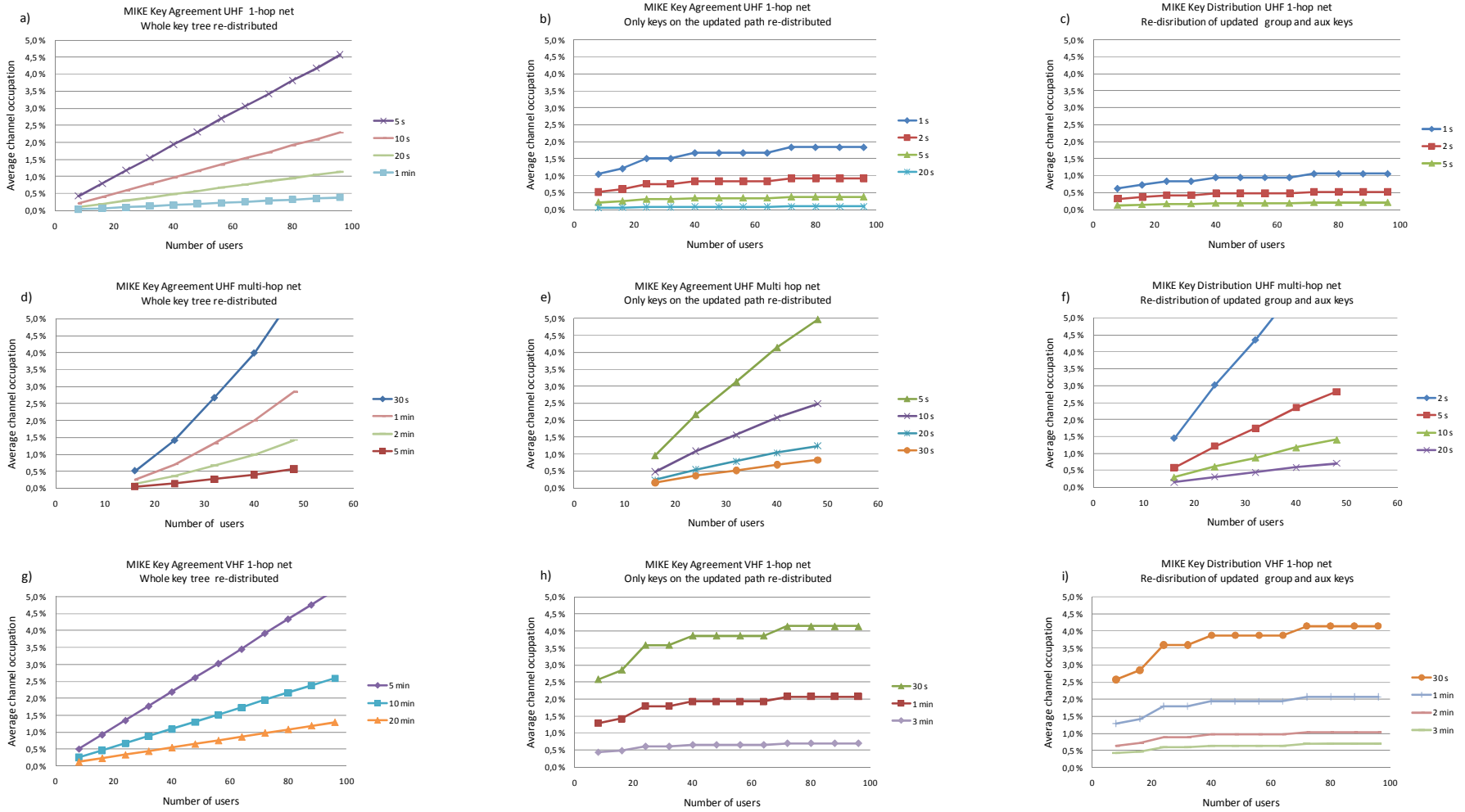



**Figure 7 Average channel occupation for different types of radio nets and varying conditions**

### 6.1.3 Allow Key overlap

Allowing the new key to co-exist with the previous one during a transition period and putting the new key into use after the message has been repeated a number of times will thus also reduce the risk of disruption and contribute to a smoother key change. At least in Key Distribution this is a viable approach.

### 6.1.4 Skip LeaveConfirm

The LeaveConfirm message appears to be superfluous. The leaving node will be able to detect its successful leave request as the KDC or TM multicasts the next UpdateDistribute message, in which it finds no new keys for itself. Nodes that intend to leave may also have left the network before the leave confirm has been received. We therefore suggest the LeaveConfirm message is skipped as illustrated in Figure 8.

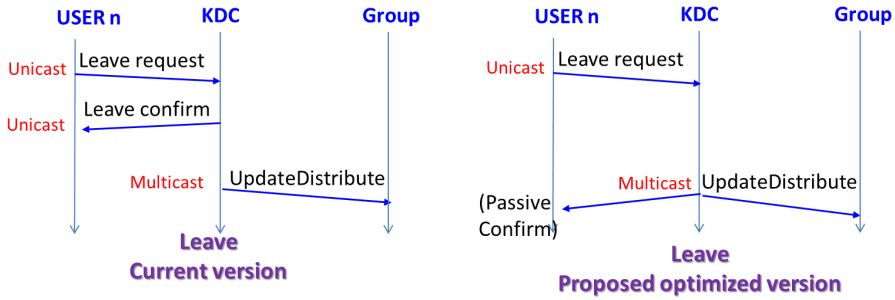



Figure 8 Proposed simplification of the leave protocol

Alternatively the LeaveConfirm message can be used as the second message in the proposed three-way handshake.

### 6.1.5 Introduce backup KDC/TM

A backup KDC/TM can to some extent reduce the problem of the central entity as a single point of failure. This is a matter of configuration rather than an optimization of the MIKE scheme. If the backup/hot standby is not co-located with the current KDC/TM, it would be beneficial if their synchronization could be handled out of band of the wireless network.

The Key Agreement mode already includes a scheme for electing a new TM when the current fails. But the election procedure includes a bandwidth consuming ring communication. Bandwidth can be saved by simply using the previous TM as a backup. If the current TM fails, the previous takes over. Only if this one also fails, the TM election scheme is used.

### 6.1.6 Distribute CRL only to the KDC

A PKI based scheme will usually require distribution of revocation information to all participants in the network. Both distribution of Certificate Revocation Lists (CRL) and on-line certificate validation have proved to be bandwidth demanding in ad-hoc networks. In the Key Distribution mode, the KDC controls which nodes remain included in the network. The CRL could therefore only be distributed (unicast) to the KDC. A drawback is that if the certificate of the KDC has been revoked, the nodes will not be made aware of this. However, compromise of the KDC compromises the system security anyway.

### 6.1.7 Change group key only on ejects

The risk of disruption due to key changes caused by Joins and Leaves can be diminished by changing the group key only when nodes need to be expelled in Key Distribution mode. The requirement for forward secrecy is still fulfilled.

### 6.1.8 Collapse TMdistribute and UpdateDistribute

After the three-way handshake, both the current TM and the new node are able to calculate the new group key as well as the blind keys on the path between the new node and the root. In the existing version of the protocol, the current TM multicasts the unaltered blind keys in the TMDistribute message. Then the new TM takes over and distributes the new blind keys in the UpdateDistribute. We suggest that these two messages are collapsed as illustrated in Figure 9 in order to save bandwidth. What is lost with this optimization is the new TM's simultaneous confirmation that it has accepted the role as new TM. But this is confirmed when the newcomer answers the next join or leave request. The old TM holds the status as backup TM until it hears the new TM answer the next request.

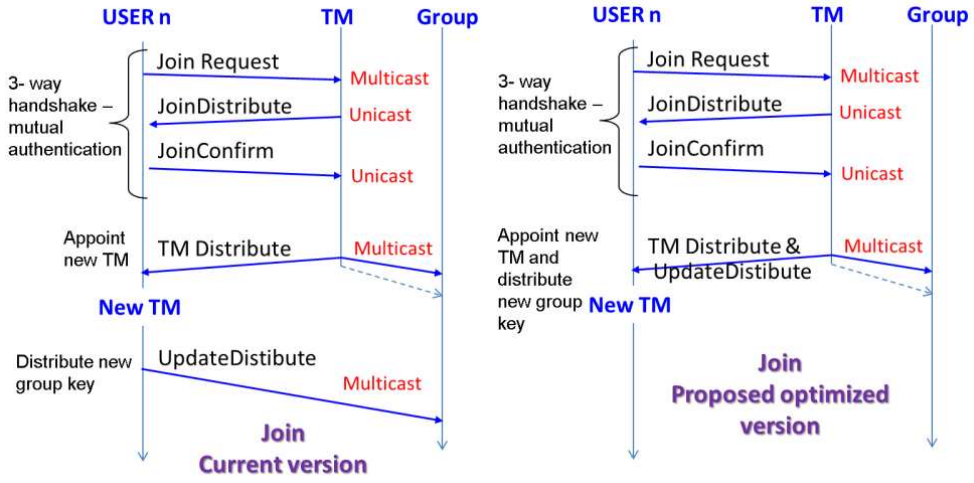



**Figure 9 Proposed Key Agreement optimization: collapse the p3TMDistribute and p3UpdateDistribute messages**

### 6.1.9 Add TM willingness

The role as TM should be left to the more powerful, protected, and well connected nodes as its tasks are resource consuming and demand good connectivity. It is therefore suggested that TM willingness is included in the three-way handshake. The current TM remains TM when a less willing node joins the network.

### 6.1.10 Do not retransmit entire key tree on Join and Leave

An optimization pointed out by T. Aurisch, is that in the current MIKE implementation the full key tree is transmitted. But only the changes need to be transmitted to the old nodes. The effect of this can be seen in Figure 7b), e) and h).

## 6.2 Other optimizations: Unbalanced key trees and batched rekeying

Reference [6] describes two possible optimization techniques; unbalanced key trees and batched rekeying.

Nodes or clusters of nodes that are likely to be revoked can be placed closer to the root in the unbalanced key tree. This reduces the number of keys that needs to be updated. In batched rekeying, member joins and leaves are collected over some period of time before rekeying. This saves bandwidth and computational cost compared to individual rekeying after each member Join or Leave request. Batched rekeying is a compromise between performance and security. It is motivated by the assumption that especially within a military environment; membership operations come in bursts. And consequently batched rekeying is beneficial.

These techniques are designed to increase the efficiency of the key tree rather than to improve the key update process [7]. But unbalanced key trees and batched rekeying also help saving bandwidth. Batched rekeying reduces the number of messages. This benefit comes at the price of delayed member Joins and Leaves, though. In the preparatory phase during network formation, this is probably acceptable. At later stages during the operation it is not. The users should not experience disruption. Re-join delays of more than a few seconds (or less) are not tolerable. When a friendly force moves into the shooting range of another group, it must start receiving and sending position data immediately.

Unbalanced key trees are beneficial for the *size* of the UpdateDistribute message. It does not reduce the *number of* messages to be sent. The root key must still be updated. Furthermore, it can be hard to decide which nodes are the most likely ones to be expelled. Consequently, this optimization is only to some extent beneficial in our scenario.

## 7. Concluding remarks

The Key Agreement mode of operations has been proposed for tactical use and Key Distribution mode for strategic networks. However, for bandwidth consumption and robustness we believe that Key Distribution is a better solution also for tactical networks. MIKE is designed for networks with good connectivity.

Main challenges for the use of MIKE Key Agreement mode in tactical mobile ad hoc networks are the requirement for a key change every time a node Joins or Leaves the group and its bandwidth consumption and delay. The forced key update on group changes leaves the Key Agreement mode vulnerable to varying connectivity. This is a problem especially for weakly connected radio nodes on the edges of the network. Nodes can be excluded accidentally by a single packet loss. Packet losses, due to varying connectivity as well as hidden and exposed nodes, are expected in our scenario. Optimizations such as repeated messages and sending only the strictly necessary keys diminish the problems only to some degree.

A remaining challenge for both modes of operation is the dependency on reliable multicast. Measures such as FEC, timeouts and periodically repetition of the latest UpdateDistribute message only to some extent reduce the problem.

A major disadvantage of all centralized key management mechanisms such as the Key Distribution mode is the existence of a single point of failure [8]. A way to remedy this is to pick a KDC (and TM) that has good connectivity and consider back-up solutions. The KDC needs more protection compared to the TM as the TM does not learn all keys.

The change of operational mode is a topic for further study. Whereas the idea of dynamic change between Key Agreement and Key Distribution mode and vice versa is interesting, we are not sure whether this provides clear benefits of a manual change. Detailed protocol specifications and a formal security analysis are also topics for further work.

A major benefit of MIKE compared to a traditional PPK w/KDC scheme is its possibility to dynamically include new group members. But as Table 1 shows, MIKE is not clearly superior to the traditional symmetric scheme.

# APPENDIX A Message formats assumed in the calculations

The MIKE protocol messages are only outlined in the existing literature. References [6], [13] and [14] give an overview of parameters such as IDs, sequence numbers, nonces, public Diffie-Hellman values and signatures that shall be included in the various messages. But the sizes of these fields and other necessary protocol parameters are not detailed. A number of assumptions were therefore made for the calculations in this report. One option was of course to reverse engineer the current implementation. Instead a theoretical approach was chosen under the assumption that a qualified discussion on necessary contents and message sizes could be an even more valuable contribution to the specification of MIKE. We have therefore made an effort to decide necessary fields and proper sizes from a theoretical point of view. This section provides the results and explains the protocol details used in the estimates.

Figure 10 outlines the assumed message formats of a MIKE Join operation. MIKE messages are encapsulated in UDP over IPv6. The MIKE message itself consists of a header and a message body. The MIKE header is needed in order to identify the type of MIKE message and information such as the length of the message body. We have assumed 8 octets will suffice. The size of the message body is variable and depends on the type of the message and key sizes. Figure 10 details the contents of the Key Distribution and Key agreement MIKE Join protocol messages inFigure 2 and Figure 4, respectively. We assume MIKE uses IPv6 addresses as identifiers, but the protocol format also allow use of other identifiers.

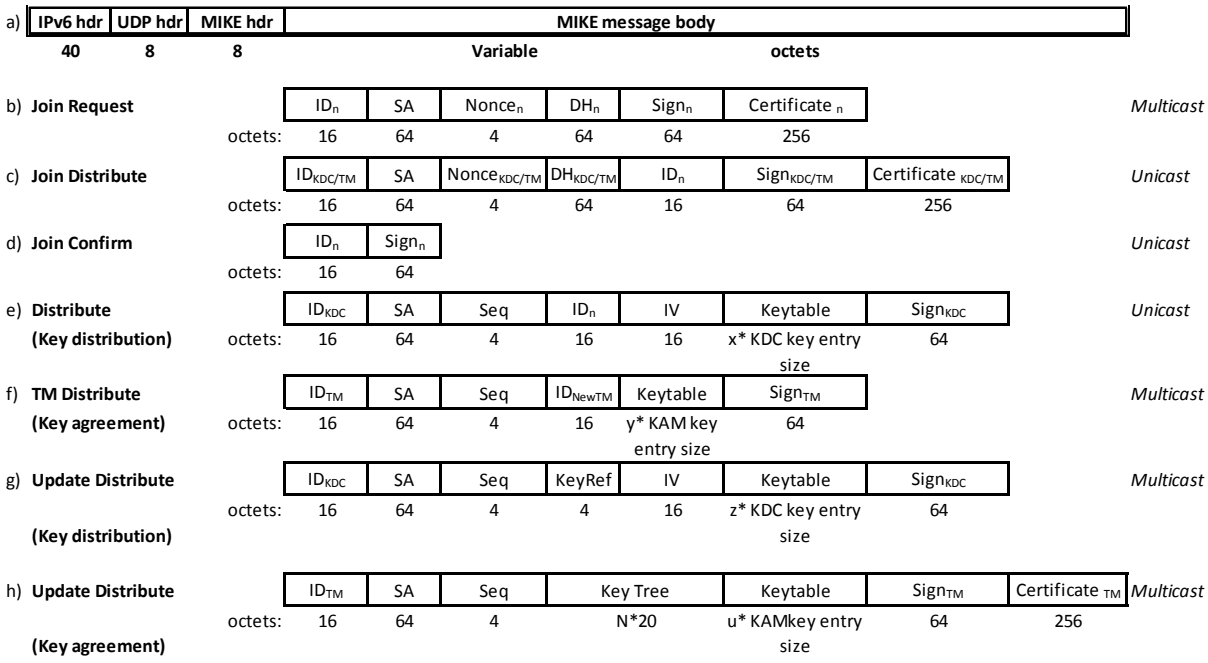



**Figure 10 MIKE encapsulation and message formats used in a Join operation**

**Join Request:** Figure 10 b) shows the assumed Join Request format. The $ID_n$ field equals the IPv6 address of the joining node n. The Security Association (SA) field must at least contain a source and a destination IPv6 address (32 bytes) plus net mask and other SA management info. The nonce field

must be large enough to avoid collision (or wrap-around in case a random nonce is replaced with a sequence number). The size of the public Diffie-Hellman value of n depends on the assumed DH scheme and the size of the generator. It is here assumed that ECDH with a group size of 256 bits is used – resulting in 512 bits = 64 octets is needed to identify the point on the elliptic curve. It is furthermore assumed an ECDSA signature scheme with 512 bits signatures is used. If the elliptic curve based scheme were replaced with for instance RSA signatures, the same level of security would demand significantly longer signatures.

The certificate of the joining node was also included. The certificate is required in order to enable the receiver of the message to verify the signature and authenticate the sender. That is, the security of MIKE depends on the certificate. The certificates are only need for the mutual authentication during in the three-way handshake. They only need to be exchanged between the joining node and the KDC/TM. The current implementation of MIKE assumes that certificates are distributed via another protocol. However, as it is a necessary prerequisite for the verification of this message and its distribution over the air consumes bandwidth, we found it reasonable to include it in the message.

Apart from the certificate field, the other fields that are included comply with the fields described in [6] and [14]. It could be argued that the ID of the joining node ($ID_n$) could be left out as the IPv6 address of the joining node is also included in the IPv6 header's source field. However, including the ID field also in the MIKE message body renders MIKE independent of identifiers at the lower layers of the protocol stack.

**Join Distribute:** Figure 10 c) illustrates the format of the Join Distribute message sent from the KDC or TM to the joining node in response to the Join Request message.

$ID_{KDC/TM}$ equals the IPv6 address of the KDC or Transaction Manager (TM). SA specifies the security association to which this message applies. The nonce and public Diffie-Hellman values returned from the KDC or TM are parallel to those sent in the previous message from the joining node.

The message format is in accordance with the specification in [6] with the addition of the certificate of the KDC/TM and the identity $ID_n$ of the joining node that the message is intended for. The certificate is needed by the joining node in order to verify the signature of this (and subsequent) message(s) from the KDC/TM.

The ID of the joining node is not strictly necessary. It has been added for completeness. The receiver is able to verify that the message is the response to its Join Request even if the $ID_n$ is omitted. This is as the signature of the Join Distribute message also includes the nonce ($Nonce_n$) of the joining node. In addition to the nonce from the joining node, the signature in $Sign_{KDC/TM}$ covers the following fields of the Join Distribute message: $ID_{KDC/TM}$, SA, $Nonce_{KDC/TM}$, $DH_{KDC/TM}$ and $ID_n$.

For simplicity, it has been assumed that all certificates have the same size (256 octets). This may not always be the case. The KDC – knowing all keys of the system – is a more powerful entity than the others. Longer keys and certificates - may be required for such instances.

**Join Confirm:** is shown in Figure 10 d). In this step the joining node completes the mutual authentication/ three-way handshake by returning a signature $Sign_n$ that covers its own ID ($ID_n$), SA, the nonces exchanged in the first two messages ($Nonce_n$ and $Nonce_{KDC/TM}$) plus the public Diffie Hellman value($DH_{KDC/TM}$)received the JoinDistribute message. This is in accordance with the description in [6]. In addition we have added the ID of the joining node to the Join Confirm message. This makes the Join Confirm message independent upon identifiers from other layers of the protocol stack. The receiving KDC/TM may potentially handle more three-way handshakes at the same time.

Including the ID of the joining node makes it easy for the KDC/TM to find out which handshake process this JoinConfirm relates to.

**Distribute (Key Distribution mode):** Figure 10 e) outlines the Distribute message used in the Key Distribution mode of operation. $ID_{KDC}$ is the IPv6 address of the KDC. SA specifies the security association this message applies to. Seq is a sequence number type of nonce used by the KDC in order to enable the recipients to detect replays. The nonce/sequence number field must be chosen large enough to prevent that it wraps around before the key is changed. If the group key changes very often, the Seq field can be smaller. The ID of the joining node ($ID_n$) has been included for the same reason as in previous messages; to decouple the MIKE protocol from lower layers' identifiers.

The Keytable and the fields prior to the IV field of the Distribute message are encrypted with the symmetric key that was established during the Diffie-Hellman key exchange in the three-way handshake. We assume an encryption algorithm such as AES and an encryption mode that demands an initialization vector (IV) are used. An IV field is therefore introduced. The Keytable entries sent to the joining node include the group key and necessary auxiliary keys. The calculations assume each key entry consists of a Key Reference (4 octets) and the key value (32octets).

The number of auxiliary keys depends on the height of the key tree. Assuming a binary key tree, the number of keys to be transferred, $x=ceil\ (log_2(N))$ where $N$ represents the number of nodes in the group (Users/leaves in the key tree).

The signature field is assumed to cover all fields in the Distribute message (apart from the signature field itself).

The $ID_n$ and IV fields are new compared to existing literature.

**TM Distribute (Key Agreement mode):** Figure 10 f) shows the format of the TM Distribute message used in Key agreement mode of operation. $ID_{TM}$ equals the IPv6 address of the current TM. SA specifies the security association this message applies to. Seq is a sequence number used to enable detection of replays. (We assume when a new node takes over as TM, it continues incrementing the sequence number used by the previous TM). The new TM is announced in the TM Distribute message. The $ID_{NewTM}$ field is therefore introduced.

We assume that the TM Distribute message is signed but not encrypted. The $Sign_{TM}$ field contains the signature calculated over all the message fields except the signature field. The Keytable field contains blinded keys. Each entry in the key table consists of the blinded key and its position. All unaltered blind keys, i.e., all blind keys *except* those on the path from the newcomer to the root, are transferred in the TM Distribute message. Each intermediate node in the key tree contains two blinded keys. The number of key entries that must be transferred in binary tree with $N$ leaf nodes (users) therefore equals $y=3N-2*ceil(log_2(N))-3$. See APPENDIX B.

**Update Distribute (Key Distribution mode):** is illustrated in Figure 10 g). $ID_{KDC}$ equals the IPv6 address of the KDC. SA specifies the security association to which this message applies. The incrementing sequence number Seq enable the recipients to detect replays of the message.

We assume each entry in the Key table includes a key reference plus the key value. The key ref enables seamless key updates and makes it unnecessary for the users to know their places in the key tree. The keys in the Keytable are encrypted with the new symmetric key established during the three-way handshake or the previous group or auxiliary keys. The key ref for the key encryption key

identifies the key that is needed to decrypt a specific entry in the Keytable. We also assume that an encryption mode that requires an initialization vector (IV).

The report assumes that the Update Distribute message contains the new group key encrypted with the previous group key plus new auxiliary keys encrypted with the previous auxiliary keys. The number of keys in the Update Distribute thus equals $z=ceil\ (log_2 N)$ where $N$ is the number of users/leaves in the key tree.

**Update Distribute (Key Agreement mode):** is illustrated in Figure 10 h). $ID_{TM}$ equals the IPv6 address of the TM. SA specifies the security association to which this message applies. The incrementing sequence number Seq enable the recipients to detect replays of the message.

The KeyTree field describes the key tree. It includes the position of each user/leaf node. Each leaf node is defined by position (row, position) and IPv6 address.

In Key Agreement mode the keys are identified by their position in the key tree. The Key table field contains key entries consisting of the blind key plus the position. Update Distribute message is assumed to carry the blind keys on the path from the joining node to the root of the tree. The number of keys is $z=2*ceil\ (log_2 N)-1$. If the total tree is distributed in the UpdateDistribute, the total number of blinded keys equals: $3N-4$. See APPENDIX B.

The certificate of the new TM is also included. This is as the TM changes, and the receivers of the Update Distribute need this certificate to verify the signature of the message. The certificate of the joining node is distributed to all in the JoinRequest multicast message. Under the assumption that the joining node always becomes the new TM, it need not be included also in the Update Distribute message. However, the inclusion makes the protocol more robust to packet losses. Nodes that did not hear the JoinRequest message will still be able to verify the Update Distribute message from the new TM. Furthermore, the inclusion of the certificate field also needed in situations where other nodes than the newcomer shall take over the TM role.

## Other assumptions

The KDC/TM was assumed to be located within the 1-hop range of the joining node. That is, the calculations do not include channel occupation caused by forwarding of unicast messages between the joining node and the KDC/TM. The calculations focus on the channel occupation as seen from the 1-hop neighborhood of the joining node.

## APPENDIX B The number of blinded keys in Key Agreement mode

Figure 11 shows a key tree used in Key Agreement mode. The User leaf at the bottom right has just joined the tree. The information transferred in the TMDistribute message originate from the nodes shown in yellow (intermediate nodes) and grey (user leafs), whereas the information transferred in the UpdateDistribute originates from the nodes shown in blue.

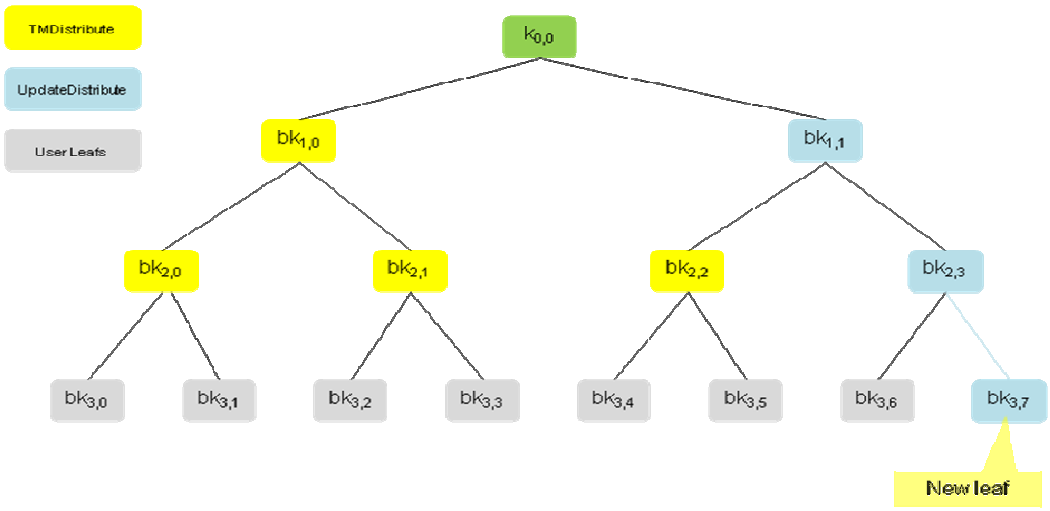



**Figure 11 Key tree for Key Agreement mode**

Note that

1. No information related to the root is transferred. The root key is derived from the blind keys of the next level.

2. Each of the intermediate nodes (Yellow and Blue) requires two blinded keys

3. The user leafs (Grey and Blue) have only one blinded key

For a general binary tree with N user leafs we see that:

*I) TMDistribute: includes all unaltered blind keys – i.e. all except those on the path from the joining node to the root.*

*a.* The total number of nodes in the key tree is: *2N-1*

*b.* Excluding the root, the number of nodes in the key tree is: *2N-2*

*c.* The total number of nodes at the intermediate level (excluding the root and leaves ) is: *2N-2-N= N-2*

*d.* The number of intermediate nodes along the path from the new leaf is: *ceiling($log_2$(N))-1*

*e.* The number of intermediate nodes affected by TMDistribute (those not on the path from the new leaf to the root):
c - d = $N-2-(ceiling(log_2(N))-1) = N-ceiling(log_2(N))-1$

*f.* The number of blind keys from each intermediate node: *2*

*g.* The number of blind keys at each user leaf: *1*

***h.*** Number of blind keys distributed in the TMDistribute message:
*1*(affected user leaf) + 2*(affected intermediate nodes) =*
$1*(N-1)+2*( N-ceiling(log_2 (N))-1) =$ ***$3N-2ceiling(log_2 (N))-3$***

*II) Update Distribute: includes blinded keys on the path from the joining leaf to the root*

*a.* The number of intermediate nodes along the path from the new leaf is:
$ceiling(log_2(N))-1$

*b.* Each intermediate node contains two blinded keys. The number of blinded keys at the intermediate nodes is therefore 2* $(ceiling(log_2(N))-1)$

*c.* The new user leaf contributes with one blinded key

*d.* The total number of blinded keys in the Update Distribute message is therefore:
2* $(ceiling(log_2(N))-1)+1=2* ceiling(log_2(N))-1$

*III) Total number of blinded keys in the key tree (the number of keys transferred in the TM Distribute + Update Distribute )):*

*a. Number of intermediate nodes: N-2 (excludes the root and the leaf level) – each containing 2 blinded keys*

*b. Number of leaf nodes: N – each containing 1 blind key*

*c. Total number of blinded keys in the key tree (excluding the root node):*
$2*(N-2)+1*N=3N-4$

# APPENDIX C Constants

| *Parameter* | *Size (octets)* |
|---|---|
| IPv6 address | 16 |
| IPv6 header | 40 |
| UDP header | 8 |
| MAC header (UHF) | 34 |
| MAC header (VHF) | 30 |
| MTU (IP layer) | 1280 |
| VHF MAC MTU | 600 |
| MIKE message header | 8 |
| Nonce | 4 |
| ID | 16 |
| SA | 64 |
| Certificate | 256 |
| Public DH | 64 |
| Sign | 64 |
| IV | 16 |
| Key Ref | 4 |
| Symmetrical key | 32 |
| Blind Key + position(3) | 67 |
| Seq number | 4 |

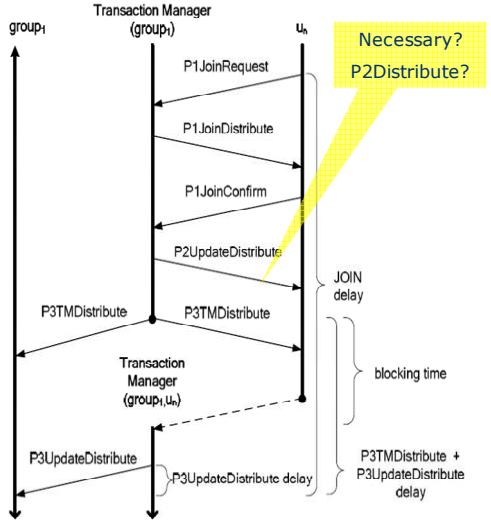
| *Parameter* | *seconds* |
|---|---|
| UHF channel access | 5,52E-04 |
| VHF channel access | 1,25E-01 |

| *Parameter* | *bit/s* |
|---|---|
| UHF Broadcast rate | 1,00E+06 |
| VHF Broadcast rate | 2,00E+04 |

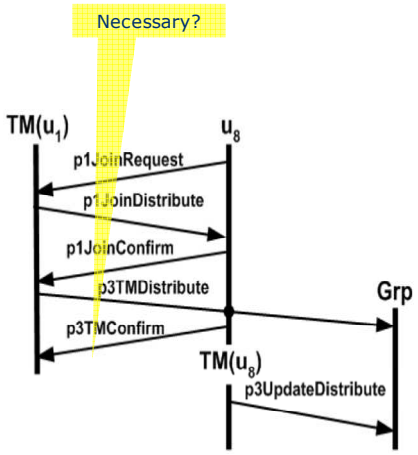
| | |
|---|---|
| FEC overhead | 20 % |

# APPENDIX D Outline of different variants of the MIKE Join Protocol

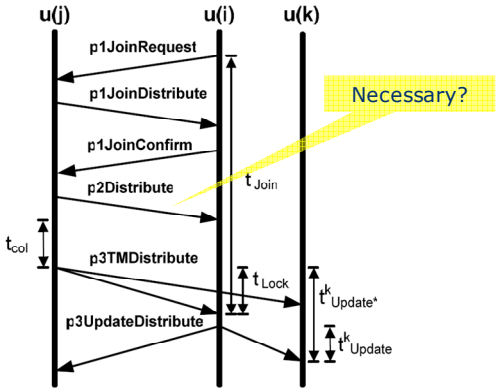
MIKE is documented through a number of publications [6][7][8][10][13][14]. The specification of the protocols varies slightly in the different publications. Figure 12 illustrates the diversity. The assessment assumes that the latest - Figure 12d) - applies. The p2Distribute, also called P2UpdateDistribute, and the p3TMConfirm are superfluous, and are omitted in this newest version.



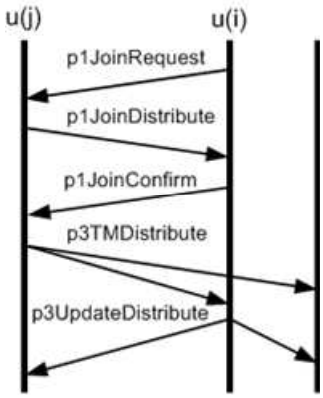

a) From: *"Practical Efficiency Analysis Of A Dual Mode Group Key Management" 2008*




b) From: *"Optimization Techniques For Military Multicast Key Management" 2005*




c) From: *"Practical Efficiency Analysis of a Tree-based Dual Mode Group Key Management" 2007*




d) From: *"System Specification of the IDP MIKE System" 2010*

**Figure 12 Protocol specification for Join in the Key Agreement mode in [8], [7], [13] and [14]**

Figure 12 a) includes a P2UpdateDistribute message, b) adds an extra p3TMConfirm message, but no P2Update, c) includes a p2Distribute, but no p3TMConfirm, d) [14] is the latest and omits both p2Distribute and p3TMConfirm.