16th ICCRTS

"Collective C2 in Multinational Civil-Military Operations"

Title of Paper

Securely connecting instant messaging systems for ad hoc networks to server based systems

Topic(s)

Networks and Networking

Name of Author(s)

Thorsten Aurisch, Philipp Steinmetz

Fraunhofer FKIE

Neuenahrer Str. 20

53343 Wachtberg, Germany

thorsten.aurisch, philipp.steinmetz@fkie.fraunhofer.de

Point of Contact

Philipp Steinmetz

Fraunhofer-Institut für Kommunikation, Informationsverarbeitung und Ergonomie FKIE

Neuenahrer Str. 20

53343 Wachtberg

Germany

+49 228 9435 593

philipp.steinmetz@fkie.fraunhofer.de

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2011		**00-00-2011 to 00-00-2011**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Securely connecting instant messaging systems for ad hoc networks to server based systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Fraunhofer FKIE,Neuenahrer Str. 20,53343 Wachtberg, Germany,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 16th International Command and Control Research and Technology Symposium (ICCRTS 2011), Qu?c City, Qu?c, Canada, June 21-23, 2011. U.S. Government or Federal Rights License.

14. ABSTRACT
Instant Messaging (IM) is a valuable tool for military users. It provides immediate text message exchange even when the available data transmission rate is low. We present a mechanism for connecting server based IM infrastructure to our IM application for tactical networks. The Chat and Instant Messaging for Tactical Environments (CIM-TE) application enables real-time text message exchange in tactical mobile ad hoc networks (MANETs). It provides secure group chat to a set of connected devices. No central server node is required. Some nodes, for example devices in vehicles, may be connected to an infrastructure IM server implementing the Extensible Messaging and Presence Protocol (XMPP). They connect the other CIM-TE devices to the XMPP server acting as protocol gateways. Thus, an XMPP client can join the CIM-TE group chat and instant messaging can be extended to tactical networks. Digital signatures ensure end-to-end integrity and authenticity of messages, while encryption provides message confidentiality. The key establishment protocol MIKE (Multicast Internet Key Exchange) provides a common encryption key for the CIM-TE nodes. If nodes join or leave the group, a new key is set automatically. The gateway functionality is the main focus of the paper.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**26**	

Abstract

Instant Messaging (IM) is a valuable tool for military users. It provides immediate text message exchange even when the available data transmission rate is low. We present a mechanism for connecting server based IM infrastructure to our IM application for tactical networks. The Chat and Instant Messaging for Tactical Environments (CIM-TE) application enables real-time text message exchange in tactical mobile ad hoc networks (MANETs). It provides secure group chat to a set of connected devices. No central server node is required. Some nodes, for example devices in vehicles, may be connected to an infrastructure IM server implementing the Extensible Messaging and Presence Protocol (XMPP). They connect the other CIM-TE devices to the XMPP server acting as protocol gateways. Thus, an XMPP client can join the CIM-TE group chat and instant messaging can be extended to tactical networks. Digital signatures ensure end-to-end integrity and authenticity of messages, while encryption provides message confidentiality. The key establishment protocol MIKE (Multicast Internet Key Exchange) provides a common encryption key for the CIM-TE nodes. If nodes join or leave the group, a new key is set automatically. The gateway functionality is the main focus of the paper.

1. Introduction

Real-time text message exchange is a popular means of communication among internet users. Since it works in low data rate environments, it is also a valuable tool for use in tactical mobile ad hoc networks. It allows users to transmit information in noisy environments or when silence is required. It has sparked interest in the military for many different applications [1][2]. Messages can be recorded by the receiving device and read later.

When designing a tactical IM, we first considered using or extending XMPP. While the XMPP standard does not require a specific architecture, it says that „it usually has been implemented via a client-server infrastructure“. Use of TCP streams, while not required either is expected throughout the standard.

In order to prevent a single point of failure we decided to design CIM-TE, a protocol without server nodes.

If there is intermittent access to a strategic network, we want the tactical users to be able to use this connection for communication with the users of such a network. This led to the development of the CIM-TE/XMPP gateway.

In this paper we first describe the CIM-TE protocol we developed for instant messaging in tactical networks. Next we describe the XMPP protocol, a popular instant messaging protocol based on a client/server architecture. Then we propose requirements for a gateway translating between these protocols, followed by a design to fulfill them. We describe our implementation of the gateway functionality and list the computational cost associated with using it. After outlining an optimization to CIM-TE we present our conclusion.

2. The CIM-TE protocol

We designed CIM-TE, a protocol for text messaging in tactical environments. It uses IP multicast to distribute messages among a group of users. Our main design goals was to provide several security features without a single point of failure, such as a server node. The protocol provides message confidentiality, authenticity, integrity and non-repudiation.

Each user of the protocol is a member of a specific group. A user's node joins the multicast address associated with this group. All messages are sent to this address and, unless transmission fails, received by all other group members.

There are two message types. Text messages are sent by users to transmit text. Presence messages are sent periodically by the nodes to indicate their status. Their absence indicates loss of connection to this user. We also use these presence messages for text message acknowledgement. The ids of recently received

messages are appended to presence messages. A text message sender retransmits unacknowledged messages within limits.
A radio silent mode can be switched on by the user, which causes the node to stop sending any messages after a final presence message is sent to notify the other users. Such a radio silent node can still receive messages, although the other users can no longer determine whether it received them or moved out of range.
Figure 1 shows a CIM-TE message. Message content is encrypted using AES (Advanced Encryption Standard). This symmetric encryption scheme requires a shared secret key among the group members. It can be either a pre-shared key or a key established by an external mechanism. We use the MIKE (Multicast Internet Key Exchange) protocol, a key establishment scheme based on key trees and Diffie-Hellman key establishment. It dynamically changes the group key whenever group members join or leave. A key identifier (KeyID) and a flag indicating the sender's status with regards to the MIKE protocol (TMFlag) are part of the CIM-TE message to facilitate MIKE usage. Details of MIKE [4] and how to secure CIM-TE with it are outside the scope of this paper.
Messages are signed using the Elliptic Curve Digital Signature Algorithm (ECDSA). Each user has a private signing key. We assume that the certificates containing a user's public key are available to the other possible members of his group. Further information on the CIM-TE protocol is available in [3].

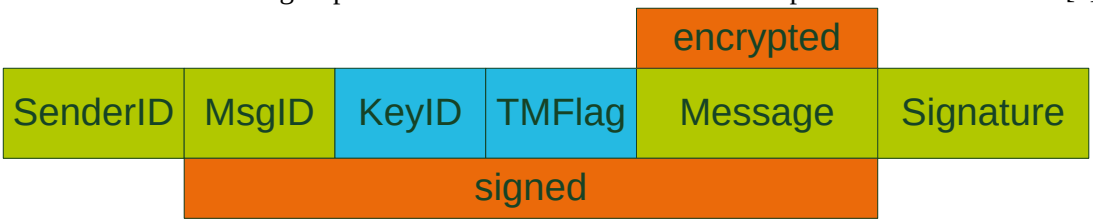


Figure 1: A CIM-TE message

3. The XMPP protocol

XMPP is a popular Instant Messaging standard specified in RFC 3920 [5] and RFC 3921 [6]. It has a client/server architecture. Each client logs into a specific server. Several servers can coexist and forward messages to each other. Users/Clients can send text messages to clients connected to either the same or to other servers. They can also receive status information about other clients such as whether they are online and short status descriptions set by the respective user.
When connecting, a client and server open a pair of XML streams, one per direction of transmission. These are used to transmit messages called stanzas to each other.
We have chosen XMPP as the protocol we want to translate CIM-TE to, because it is a well documented and extensible standard. It has also become popular among military users such as the US DoD and NATO. XMPP recommends use of Transport Layer Security (TLS) to provide confidentiality and of Simple Authentication and Security Layer (SASL) to provide authenticity. In this paper we assume that both are in use when discussing XMPP.

4. CIM-TE/XMPP gateway scenario and requirements

As an example for gateway usage we consider several dismounted soldiers equipped with handheld devices and a vehicle with a more powerful antenna accompanying them. The handhelds and the device on the vehicle form a CIM-TE group. When the vehicle is able to connect to a network at an HQ with an XMPP server, it can login, connect to a user and share this connection with the dismounted soldiers. This allows the user at the HQ to provide new information, orders, etc. to the whole group. Figure 2 shows a gateway on a vehicle connecting both networks.

We have a CIM-TE node connected both to an infrastructure network and to other CIM-TE nodes not connected to this infrastructure network. An XMPP server is connected to the infrastructure network. We want the node connected to both networks to act as a gateway to provide the other nodes with access to the XMPP server.
The gateway node has to maintain the security features of CIM-TE and XMPP. Message confidentiality, authenticity, integrity and non-repudiation have to be provided.

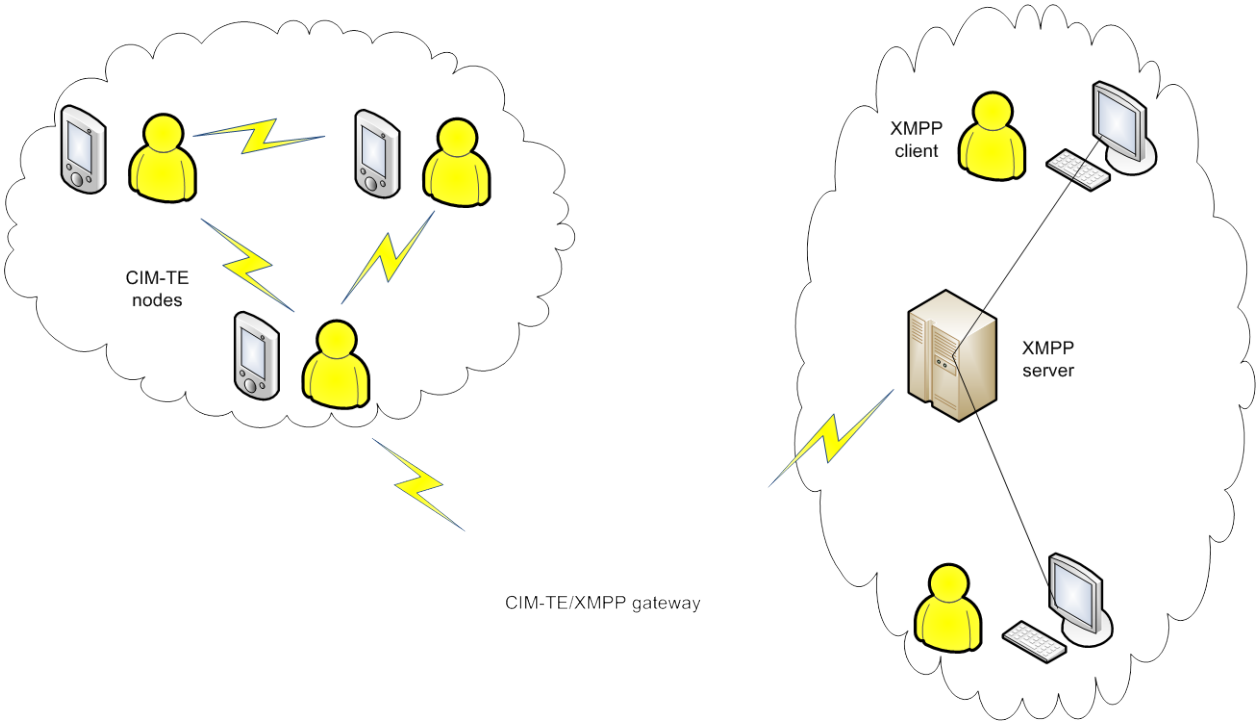


Figure 2: A CIM-TE/XMPP gateway

5. CIM-TE/XMPP gateway design

Two RFCs provide important input to the design of such a gateway. RFC 3922 [8] specifies a mapping from XMPP to Common Presence and Instant Messaging (CPIM). RFC 3923 [9] specifies end-to-end signing and encryption for XMPP.
As described in these RFCs we use the CPIM format for text messages and the PIDF format for presence messages. These elements are signed using a digital signature scheme. As in CIM-TE we use the Elliptic Curve Digital Signature Algorithm (ECDSA) because of its efficiency [12]. Both CIM-TE packets and XMPP stanzas contain these signed elements. Figure 3 shows a CIM-TE message containing a signed CPIM element in place of a simple text payload. When messages are translated from one protocol to the other at the gateway, these elements are passed without being modified. This way the endpoints can verify the signatures of the original senders. This provides message integrity, authenticity and non-repudiation, since only the original sender has access to the private signing key.
While other authentication mechanisms are more popular among civilian XMPP users, we consider the necessary Public Key Infrastructure (PKI) a sensible way to handle user authentication in the military domain.
As noted above message confidentiality is achieved by CIM-TE using symmetric keys provided by MIKE. MIKE also employs digital signatures for authenticating users before providing these symmetric keys. Once the certificates of all devices meant to communicate with each other are installed on each device, no further out-of-band distribution of key material is necessary. If a user's signing key is compromised,

messages seemingly originating from this user can be generated by the attacker. The corresponding certificate has to be revoked.
XMPP uses TLS (recommended) to protect the XML streams between client and server. The gateway decrypts the incoming messages from both CIM-TE and XMPP and encrypts them when passing them to the other protocol. These symmetric reencryption operations require few computational resources. There is no unencrypted message text outside the devices.

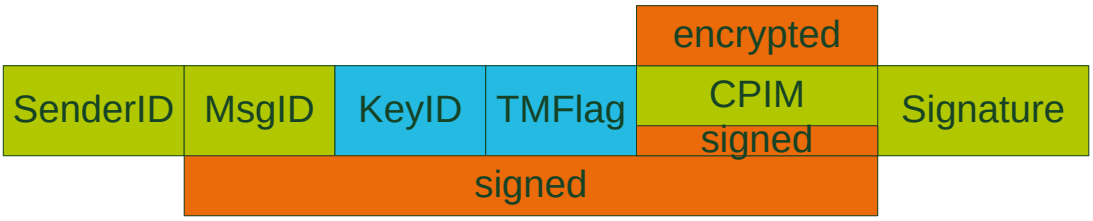


Figure 3: A CIM-TE message with CPIM payload

6. Gateway node security

The gateway node can read all messages, because it decrypts them when converting them from one protocol to the other. In our scenario the gateway node is part of the CIM-TE group which makes it an intended receiver anyway. It cannot forge messages, since it has only access to its own signing key. It can silently discard messages. It can delay and duplicate messages within the limits imposed by the timestamp mechanism. RFC 3923 recommends that a user is warned if the timestamp of a message differs from the current time by more than 5 minutes or if it is older than a valid message received in the last 10 minutes.

7. Gateway placement and usage

Since the gateway contains a CIM-TE node it requires IP multicast routing to all other CIM-TE nodes. The gateway node does not forward its own messages to prevent loops. The user of the gateway node is responsible for providing login credentials to connect to the XMPP server and setting the XMPP user to be contacted. This could even be another gateway, thereby connecting two CIM-TE groups over an XMPP server.

8. Implementation

We have added the gateway functionality described here to our CIM-TE implementation (Figure 4). While earlier versions of CIM-TE were developed in Java ME to show that it runs on 400 MHz PDAs, we use Java SE for the version including gateway functionality. This allows us to use additional libraries. The Smack library provides an XMPP client and the means to add extensions such as the end-to-end signing mechanism. The Bouncy Castle library provides cryptographic functions. JavaMail is used for CPIM format and PIDF format data.
We use the Openfire Server XMPP server for testing.
Few if any civilian XMPP clients support end-to-end signing as defined in RFC 3923 and used here. Most support alternative mechanisms such as OpenPGP. This is documented in [10]. We use our own gateway implementation as XMPP client.

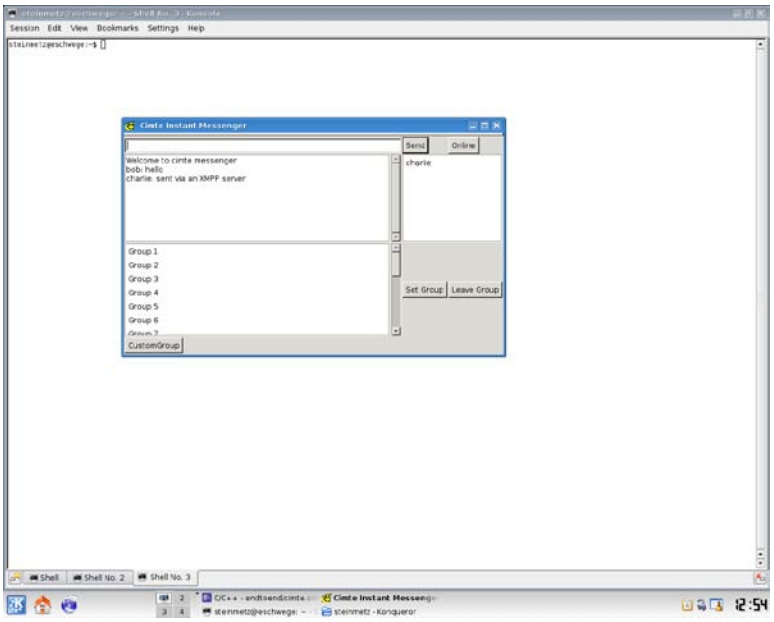


Figure 4: CIM-TE screenshot

9. Cost of cryptographic operations

Securing the messages causes computational and message size overhead. Generation of a signature by the sender and its verification by the receiver are especially relevant here. They are asymmetric cryptographic operations which require significantly more computation time than symmetric operations. In comparison, the cost of the symmetric encryption operations used for providing message confidentiality can be neglected.

A CIM-TE node has to do two signing operations when sending a message, signing both the CPIM element and the full CIM-TE message. Correspondingly it does two verification operations when receiving a message.

An XMPP node does one signing operation, signing a CPIM element, when sending a message. Since TLS uses a keyed MAC for ensuring message integrity, no additional signing operations are necessary. One verification operation is done when receiving a message.

A CIM-TE/XMPP gateway doubling as a regular node which displays the messages in addition to forwarding them has to do the work of a receiving node and the work of a sending node less signing the CPIM element. This means 2 verification operations for CIM-TE to XMPP translation and one verification and one signing operation for XMPP to CIM-TE translation (Table 1).

Regarding message size, again the signatures have to be taken into account. A CIM-TE message contains two signatures, protecting the whole message and the CPIM element. An XMPP message contains the CPIM element signature. For fixed algorithm and key size signatures have constant length independent of message length. When using a 160 bit elliptic curve, a signature requires 320 bit. This means 40 Bytes for an XMPP stanza and 80 Bytes total for a CIM-TE message. When transmitting short text messages they are relevant regarding total message size. The symmetric encryption algorithm we use, AES in CBC (Cipher Block Chaining) mode, causes minor message size increase through padding to full 16 Byte blocks.

Table 1: Computational cost of message transmission

Node activity	Signing operations	Verification operations
CIM-TE send	2	
CIM-TE receive		2
XMPP send	1	

XMPP receive		1
CIM-TE to XMPP		2
XMPP to CIM-TE	1	1

10. Optional XMPP extensions

XMPP stanzas can be extended by defining optional child elements. We discuss the relevance of two of these extensions for our gateway.

End-to-end-signing

The <e2e> child element is used by the end-to-end signing mechanism. It contains the signed CPIM or PIDF object. Both the sending and receiving client have to support this element to make the signing mechanism work. The XMPP server does not have to understand it as long as passes it to the recipient unmodified as is recommended by the RFC.
If we want to provide compatibility with clients which do not understand this extension, we can transmit the message both signed as described above and in the standard message body. This way, all clients can display the message, while only those which support the extension can verify it.

Group chat

The XMPP extension “Multi-User Chat” [11] describes communication between more than two participants. Since it does not define integration with end-to-end signing, our gateway implementation connects to a single XMPP user instead. We consider an extension of the end-to-end signing mechanism and of our gateway to support multi-user chat to be straightforward, as long as exactly one gateway is used or a mechanism for preventing message loops is introduced.

11. Optimizing CIM-TE usage

CIM-TE has originally been designed to contain a text payload instead of a CPIM element. Since we now have an additional signature providing authenticity of the element, we can remove the CIM-TE message signature and use a keyed MAC for providing authenticity of the message parts outside the CPIM element (Figure 5). MACs require significantly less computation than signatures. Unlike signatures they require a shared secret among the legitimate sender and the receiver of a message. When using a keyed MAC, a symmetric key has to be distributed to all group members, so each of them can verify that messages originate from another group member. They do not provide the means to determine which of the group members sent a message.
Unlike the encryption key the MAC key cannot be provided by MIKE, as it is intended to protect the flags used by MIKE for encryption key establishment. The MAC key has to be pre-distributed to all devices involved. Its compromise has less impact than compromise of a signature or a MIKE key. In case of compromise of the MAC key, an attacker could modify messages by changing the MsgID, KeyID and TMFlag values while keeping a valid CPIM element and computing the MAC. This could degrade availability by confusing the MIKE protocol and the message retransmission mechanism. It does not allow an attacker to generate new CPIM elements containing message text. As before, an attacker requires a user’s private signing key in order to send messages in his name.

Figure 5: A CIM-TE message with keyed MAC

This modification reduces the number of asymmetric operations when sending and receiving a CIM-TE message by one (Table 2). We also reduce CIM-TE message size by one signature length.

Table 2: Modified cost of message transmission

Node activity	Signing operations	Verification operations
CIM-TE send	1	
CIM-TE receive		1
XMPP send	1	
XMPP receive		1
CIM-TE to XMPP		1
XMPP to CIM-TE		1

12. Conclusion

We have presented a gateway between a distributed and a client/server instant messaging protocol. This is possible without sacrificing security. We have described the computational cost of maintaining message authenticity while translating between the protocols. We have implemented the gateway and described a modification to our instant messaging protocol for increasing efficiency.

References

[1] N. Heacox, R. Moore, J. Morrison, R. Yturralde, „Real-time Online ‚Chat' Use in Navy Operations", Pacific Science & Engineering Group Inc. and SPAWAR Systems Center San Diego, 2004
[2] B. Eovito, „The impact of synchronous text-based chat on military command and control", JC4I Graduate School of Operational & Informational Sciences, 2006
[3] J. Tölle, P. Steinmetz, T. Ginzler, A protocol for secure instant messaging in tactical networks, Military CIS Conference 2008, Cracow
[4] T. Aurisch, "Using key trees for securing military multicast communication", Unclassified Proceedings of the IEEE Milcom 2004, 2004
[5] P. Saint-Andre (Ed.), RFC 3920, „Extensible Messaging and Presence Protocol (XMPP): Core", Jabber Software Foundation, 2004
[6] P. Saint-Andre (Ed.), RFC 3921, „Extensible Messaging and Presence Protocol (XMPP):
[7] Instant Messaging and Presence", Jabber Software Foundation, 2004
[8] P. Saint-Andre (Ed.), RFC 3922, „Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)", Jabber Software Foundation, 2004
[9] P. Saint-Andre (Ed.), RFC 3923, „End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)", Jabber Software Foundation, 2004

[10] XMPP Standards Foundation, XEP-0027, „Current Jabber OpenPGP Usage“, http://xmpp.org/extensions
[11] XMPP Standards Foundation, XEP-0045, „Multi-User Chat“, http://xmpp.org/extensions
[12] National Security Agency, „The Case for Elliptic Curve Cryptography“, http://www.nsa.gov/business/programs/elliptic_curve.shtml

SECURELY CONNECTING INSTANT MESSAGING SYSTEMS FOR AD HOC NETWORKS TO SERVER BASED SYSTEMS

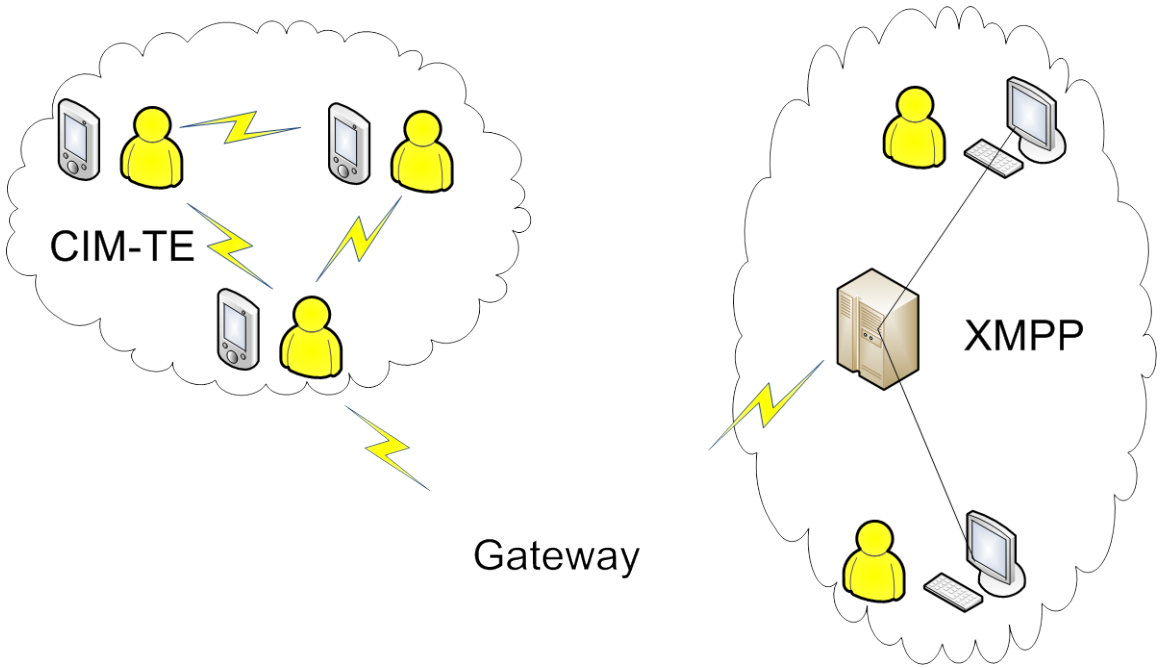
Philipp Steinmetz

Introduction

- Motivation for Tactical Instant Messaging
  - Bandwidth-efficient
  - Silent information exchange
  - Message history
- Motivation for connection to strategic networks
  - Connection to commanders and technical specialists at HQ

Goals

- Provide instant messaging in tactical networks
- Connect the tactical instant messaging protocol to XMPP



Requirements

- Tactical Instant Messaging requirements
  - Distributed system without a server
  - Security
  - Efficiency

The CIM-TE protocol

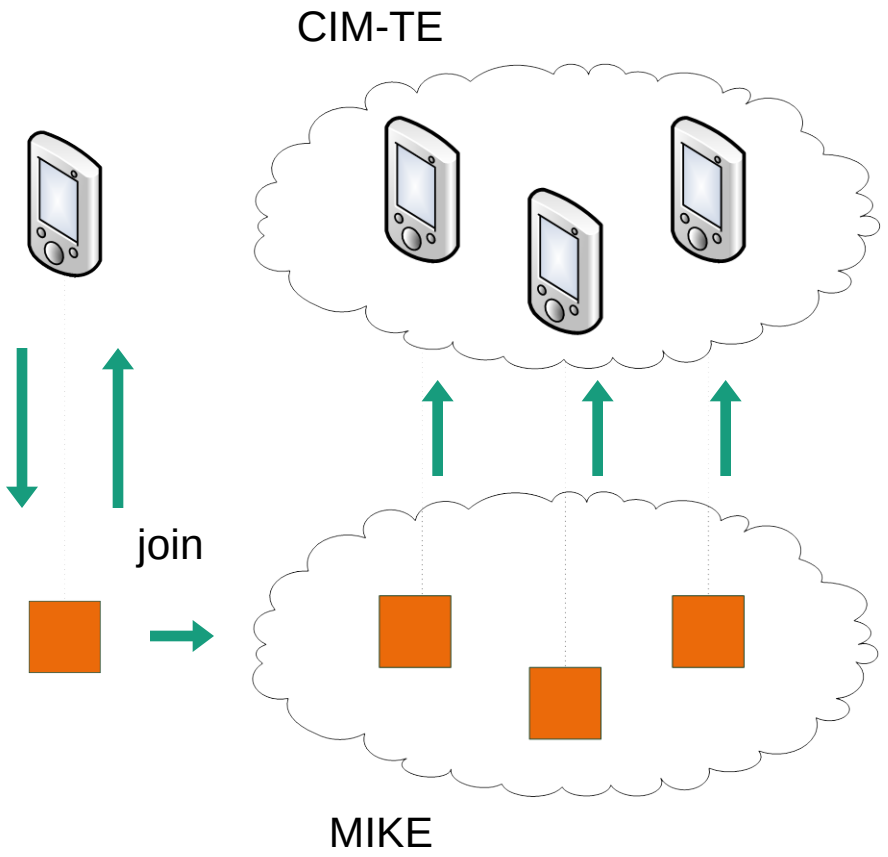
- Instant Messaging protocol for tactical environments
- Distributed system
- Uses IP multicast to distribute messages among a group of users
- Text messages
- Presence messages: status updates

The CIM-TE protocol

- Signature: Digital signature to provide authenticity
- Message: Symmetric message encrypted with AES
- SenderID, MsgID: Unique message identification
- KeyID, TMFlag: Used by MIKE protocol

The MIKE protocol

- The encryption key is provided by the MIKE protocol
- MIKE: Group key distribution protocol based on Diffie-Hellman key exchange
- Key is provided to all group members
- Key changes when members join or leave



Fraunhofer FKIE

The XMPP protocol

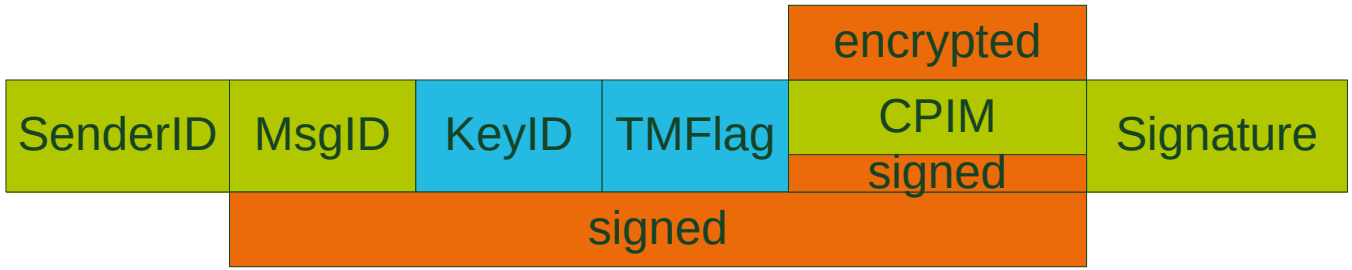
- Used in strategic networks
- Popular instant messaging standard
- XML streams between client and server
- Standards process for extensions (XEPs)

CIM-TE/XMPP gateway requirements

- Connect tactical and strategic messaging
- Maintain security features
  - confidentiality, authenticity, integrity, non-repudiation
- Limit effects of malicious gateway

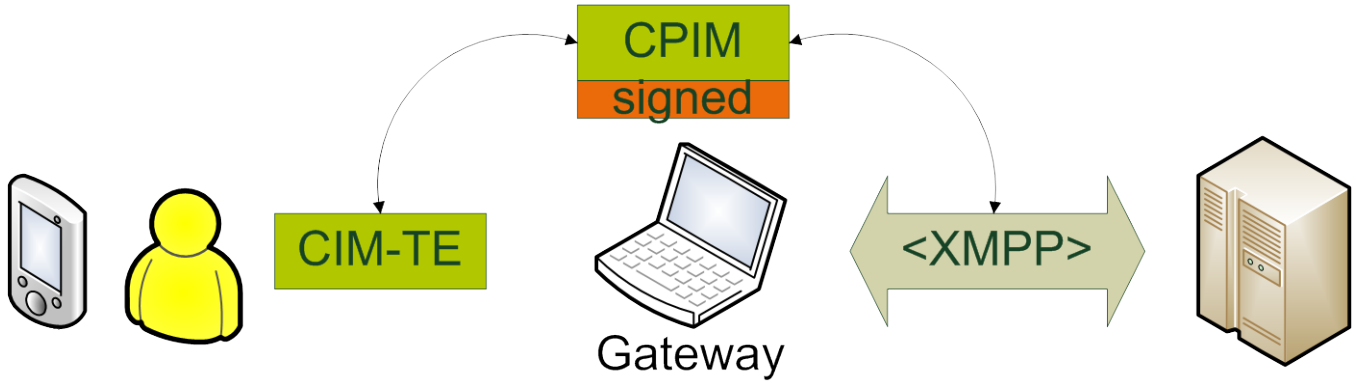
CIM-TE/XMPP gateway

- Mechanism described in RFC 3923 “End-to-end signing and object encryption for XMPP“
- Sender generates signed messages in CPIM format
- Both protocols use CPIM elements instead of plain text



CIM-TE/XMPP gateway

- Gateway translates between CIM-TE messages and XMPP XML stanzas without modifying the CPIM elements
- Receiver can verify the original sender's signature
- Gateway cannot forge text messages



CIM-TE/XMPP gateway

- Symmetric decryption and re-encryption at gateway
- Gateway is group member: access to all messages anyway
- Low computational cost for encryption
- Independent key management for each protocol

Implementation

- CIM-TE
  - Implemented in Java ME
- CIM-TE with XMPP gateway functionality
  - Implemented in Java SE
- Java advantages
  - Runs on PDAs
  - Libraries available for crypto operations, XML and XMPP

Optimization

- Possible CIM-TE optimization:
  - Replace CIM-TE signature with keyed MAC, since it contains a signed CPIM element
  - Use pre-distributed MAC key
- Message content is still signed by the sender
- Message flags are protected

Cryptographic operations

- Asymmetric crypto operations (signing, verification) are expensive
- Optimization (orange) reduces them for CIM-TE and gateway nodes

Node activity	Signing operations	Verification operations
CIM-TE send	2 1	
CIM-TE receive		2 1
XMPP send	1	
XMPP receive		1
CIM-TE to XMPP		2 1
XMPP to CIM-TE	1 0	1

Summary

- Gateway between XMPP and CIM-TE, our tactical IM
- Gateway maintains security features with end-to-end signing and re-encryption
- Java implementation

Thank you!

philipp.steinmetz@fkie.fraunhofer.de