Using MIKEY-SAKKE

Building secure multimedia services

One of the roles of CESG is to provide advice to UK government entities and organisations providing services to the UK government. The Guidance found here is provided and intended for use by this audience, i.e. UK government entities and organisations providing services to these.

CESG recognise that other organisations other than its intended audience may have access to this Guidance, in particular through being able to access this on the CESG website. However, these other organisations, whilst they may find aspects of the Guidance useful, should bear in mind that the Guidance is not specifically intended for them and that CESG’s role is not to provide advice to them.

The aim of this white paper is to provide thoughts on how a secure multimedia service may be built. These thoughts should be viewed merely as suggestions, and not a mandate. Developers are encouraged to modify or enhance the ideas presented in this white paper and build outstanding multimedia security services for various enterprise sectors, including government.

The copyright of this document is reserved and vested in the Crown.
Using MIKEY-SAKKE
Building secure multimedia services (v1.0, April 2014)

What is MIKEY-SAKKE?
MIKEY-SAKKE is a protocol designed for enterprise and government enterprise to enable secure, cross-platform multimedia communications. It is highly scalable, requiring no prior setup between users or distribution of user certificates. It is highly flexible, supporting both real-time communications (such as voice), conference calls, and deferred delivery (such as messaging and voicemail). It designed to be centrally-managed, giving a domain manager full control of the security of the system. But even so, it maintains high-availability, as calling does not require interaction with centralized architecture.

A Valuable Market
Secure communications are needed across government and to this end government has a policy of encouraging the development of security solutions. The vision is that every sensitive call, email or file-transfer in UK government and industry should be appropriately protected, and furthermore that this protection should enhance, not hinder, collaboration.

In the UK, the cross-sector secure voice market is estimated at 2 million high-value subscribers, over 3% of the subscriber base.

The requirement for secure multimedia communications exists across a variety of sectors including central government, public safety, banking, health, power and other critical national infrastructure. In the UK, the cross-sector secure voice market is estimated at 2 million high-value subscribers\(^1\), over 3% of the subscriber base. While each sector brings its own service requirements, every sector increasingly requires secure cross-sector collaboration.

There have existed some small scale, isolated, security solutions within each of these sectors for some time. However, there are few usable, scalable, regulatory compliant and above all affordable options that make enterprise deployment of multimedia security a reality for the end user. Without the scale of cross-sector interoperability, these security solutions will remain niche; or limited to a small number of end-points. The wider need will remain unfulfilled.

The Economies of Scale
Fundamentally, the success of all communication solutions are intrinsically tied to scale: The more people that you are able to communicate with, the more valuable the offering, and the greater the demand for the communication solution. It is undeniable that a wide-scale demand exists. The use of a particular secure mobile email solution is almost ubiquitous in the UK government, and frequently there will be associated sensitive conversations that would benefit from security. As data becomes less transient, the need for security increases.

“A consistent, cross-platform security solution is needed that can support messaging, voice, and video, alongside sector specific services.”

It is evident that security solutions must evolve from the small scale solutions delivered in the past. A consistent, cross-platform security solution is needed that can support messaging, voice, and video, alongside sector specific services.

In 2009, one of the barriers to commercial deployment was the lack of a viable security solution. Creating a secure, highly-scalable interconnected network, where any user can securely communicate with another user is non-trivial, and while the technology existed, there were no appropriate supporting standards.

Hence in 2010, CESG, the National Technical Authority for Information Assurance in the UK, developed a standard that could meet the need. Using a cryptographic solution developed by two Japanese cryptographers, Sakai and Kasahara, CESG standardized MIKEY-SAKKE protocol in IETF RFCs 6507 [1], 6508 [2] and 6509 [3].

---

\(^1\) 500K users in central government, 500K in health, 300K in public safety, 100K in Finance, 600K other sectors (e.g. power, law, telecoms, business services etc.).
An Evolution towards Security

It is not expected that a market of secure multimedia solutions serving the whole of government and industry will appear overnight.

The market currently contains a number of ‘early-adopters’ who have built secure islands of connected users in specific sectors using MIKEY-SAKKE.

In the near-future it is anticipated that the number of providers will increase, and accelerate as users adopt 4G. Users will have a choice of sector-specific user clients and central management systems. Gateways will also be deployed to allow these secure islands to connect to existing systems.

The long term intent is for these islands to join seamlessly to allow secure cross-government and industry communication. Naturally, the only way this will be possible is if every sector takes the same approach to the underlying security.

As a result, CESG will only certify secure Voice-Over-IP (VoIP) clients which include the MIKEY-SAKKE protocol for OFFICIAL use by UK Government. The evaluation standard is defined in [4] and [5].

Scope of this Paper

As of 2014, the MIKEY-SAKKE protocol has been standardized for 2 years and a variety of products have been, and are being, developed with the protocol at their core.

However, just as the security protocol IPsec doesn’t explain how to build a cross-site enterprise network, and the TLS protocol doesn’t explain how to build an e-commerce site, the MIKEY-SAKKE RFCs do not explain how to practically build a secure multimedia service.

Innovation is central to the success of multimedia services, both in terms of functionality and security features. That is why, beyond the details in the MIKEY-SAKKE RFCs, no further requirements have been placed upon developers that may limit ingenuity.

IPR & Source Code

CESG developed, and owns the intellectual property in, the MIKEY-SAKKE protocol. The aim was to produce an unencumbered, free-to-use protocol, which would enable a market of secure multimedia products to develop.

CESG’s understanding is that MIKEY-SAKKE can be developed, used and sold without any licensing cost, and a number of companies are already doing so. To encourage adoption, MIKEY-SAKKE developers have made a library freely available under an open source (LGPL) license [6].

Status of Components

CESG chose components for the protocol which it understood to be unencumbered when used in accordance with the RFCs. MIKEY-SAKKE comprises three components: Multimedia Internet Keying Protocol (MIKEY), Sakai-Kasahara Key Encryption (SAKKE) and Elliptic Curve-Based Certificateless Signatures for ID-based encryption (ECCSI) of which:

- The MIKEY Protocol is standardized by the IETF in RFC 3830 [7]. It is widely used and CESG has not been made aware of any patent or other IPR which would encumber the MIKEY protocol.

- Sakai-Kasahara published their ID-based cryptosystem in 2003. CESG has not been made aware of any patent or other IPR which would encumber the Sakai-Kasahara scheme. The scheme was standardized by IEEE in P1363.3 and is often referred to as being patent free (for example [8]).

- ECCSI is a minor adjustment to ECDSA. Certicom has asserted its ownership of patents relating to this scheme, and has stated that it would not enforce any rights in respect of any patents which it owns or controls against any party which makes, uses, imports, sells, or offers to sell, any product that implements the ECCSI RFC 6507 [9]. The full statement is at [9]. CESG has not been made aware of any other patent or other IPR which would encumber the protocol.
A Secure Call

To introduce the protocol, it is simplest to explain the call setup process for a secure VoIP call. A similar process can be applied to a variety of other applications including instant messaging, email and video.

Summary

Imagine Barry wishes to call Paul. When Barry calls Paul, Barry’s handset creates a key and securely sends it to Paul. Paul’s handset extracts this key, and Barry and Paul’s handsets use this key to secure the call.

An important aspect of the protocol is that Paul knows that only Barry could have sent the key and Barry knows that only Paul could extract it. As a result, the call is secured against interception.

How does this work?

VoIP calls are frequently initiated with a SIP (Session Initiation Protocol) INVITE message, often carrying SDP (Session Description Protocol) content. The SDP content can contain a MIKEY (Multimedia Internet Keying) message providing the necessary keying material.

The RFCs 6507-6509 define a type of MIKEY message containing a SAKKE cryptogram, hence the name MIKEY-SAKKE. This encapsulates a secret key, protecting it from everyone but the specific user for which it is intended. This simple mechanism allows the keying process to complete with just a single message.

Further SIP messages may be exchanged after this point, such as a SIP_OK response. This does not need to contain security information, but may optionally return a MIKEY-SAKKE I_MESSAGE to confirm successful setup.

The shared key is used to key the SRTP protocol. This secures the call during transmission.

For further details, CESG’s secure voice profile is described in [4].

A Variety of Uses

The above example, which supports voice and video calls, provides just one of many applications.

Voicemail

As it is a ‘one-way’ protocol, it can be easily adapted to voicemail/videomail. The call is redirected to the voicemail server which simply stores the SDP message and SRTP content. The voicemail server has no access to the content of the call.

Conference Call

By sending out multiple SIP INVITE messages, a user can send the same key to multiple participants, thereby beginning a secure conference call.

For group calls in a public-safety context see the section on ‘Sector-Specific Services’.

Instant Messaging

As with RTP sessions, a MIKEY-SAKKE I_MESSAGE can be provided in the SIP INVITE of an MSRP [10] session. The key provided in the I_MESSAGE can be used to key a TLS-PSK tunnel [11] between the two end-points, thereby securing the MSRP communication.

Other (e.g. email, file transfer)

For asynchronous communications, there are a number of protocol solutions. These solutions will generally involve a MIKEY-SAKKE I_MESSAGE, followed by encrypted content.

WebRTC for secure VoIP?

WebRTC is a Google standard that can be used to secure VoIP, but it is fundamentally different in design. WebRTC is primarily designed as an ad-hoc browser-based service for anonymous calls (e.g. user to advertiser), whereas MIKEY-SAKKE is an enterprise-level solution where anonymity is not possible. With MIKEY-SAKKE you always know who you are talking to.
The MIKEY-SAKKE protocol uses identity-based cryptography. To explain, as a thought-experiment, imagine a trusted domain manager provides a domain certificate giving any user the ability to take an input ‘identity’ and build a padlock based on that ‘identity’.

Also imagine that no one, except the domain manager, can create a key to open any padlock.

The domain manager securely provides the ‘Barry’ key (the key that opens the ‘Barry’ padlock) to the user verified as Barry, along with the domain’s public certificate. Now Barry can unlock a padlock, or MIKEY-SAKKE I_MESSAGE sent to his identity. In a similar way, he can sign information using his identity.

As a result, anyone can securely communicate with any user in the domain without any prior information from that user. In truth, the user need not even be online. This makes MIKEY-SAKKE highly flexible and scalable.

The technical aspects of the MIKEY-SAKKE solution are achieved using the identity-based public-key cryptographic scheme ‘Sakai-Kasahara’ (2003) [12]. This scheme is a well-studied and well-known identity-based cryptographic scheme. A security proof was produced in 2005 [13].

Identities

The cryptography of MIKEY-SAKKE is based around the use of centrally verified identities. The format of the identity will vary depending on application. It may be a telephone number, an email address, a URL, an IMPU (for VoLTE, see [14]), an IPv6 address, and so on. So long as it is known by the users who are communicating, and can be verified by a domain manager, it is a potential option for a MIKEY-SAKKE identity.

Of course, users or user equipment can simultaneously have multiple identities verified by the domain manager. For ultimate flexibility, users can even ask for new identities to be verified and keyed by the domain manager after receiving a message encrypted to the new identity.

Users can exist within multiple domains too. For example, a police constable could have the same identity within a ‘Police Domain’, a ‘Public-Safety Domain’ and a ‘Government Domain’, with each domain staying independent and maintaining independent security policies.

Taking the time

In the majority of use-cases, the domain manager will wish to add a temporal component to the identity used for user encryption. This will limit the period a user can use the identity, or should the user be compromised, the time an attacker might have access.

This also determines how frequently end-user clients will need to communicate with the domain manager. It has been suggested that identities should contain the current month. In this case, the client will need to contact the domain manager on at least a monthly basis to obtain the key for the next month’s identity.

The full string used by the MIKEY-SAKKE protocol is known as the MIKEY-SAKKE UID. It is recommended that the UID format should be determined by the domain manager and recorded in the domain’s public certificate. For example, a domain certificate may define the following UID format:

‘#uri?uidyear=#year&uidmonth=#month’

In this case, to compose a UID and communicate with a user, the ‘#uri’, ‘#year’ and ‘#month’ would be replaced:

'user@domain.com?uidyear=2014&uidmonth=03'

Where are the certificates?

MIKEY-SAKKE users do not need their own certificate. Secure communication is possible only knowing the public identity of who you want to talk to (e.g. email address or telephone number).

The domain manager has a single public certificate for the entire domain. This is required to communicate with users in the domain.
Communicating with the Domain Manager

The 'root-of-trust' of a particular domain lies with the domain manager. The domain manager is responsible for periodically (e.g. monthly) creating user keys for user identities and distributing these keys to verified users. To do this, it is anticipated that the domain manager will use a key management server (KMS).

As has already been discussed, the user’s clients will need to contact the KMS on an occasional, such as monthly, basis. As a result the KMS need not be a highly available server. Indeed it may only be online when provisioning the latest key material.

It is essential for user clients to be able to easily and securely communicate with the KMS. There has already been innovation in how this process could be implemented. In this section, a simple mechanism is provided.

KMS as a web server

Let’s consider the KMS to be a standard web-server. As with the majority of websites today, the user connects to the web server via a HTTPS session in a browser and logs in using a username and password2.

The user’s device can now request the appropriate identities (e.g. via HTTP GET requests), and receive the keying material within signed content (e.g. XML) inside the secure responses.

It is recommended that for subsequent periodic connections to the KMS, logins, identity requests and key downloads are performed by the user’s device automatically.

Innovation in the connection

The above process is just one approach. In some contexts, such as where there is a device management service, there will be no need for a username/password combination as the address of the KMS and login credentials can be pre-provisioned. Ideally, services will evolve to simply require domain managers to select a set of options determining how secure multimedia services are used within the domain, and keying will be performed automatically.

The RFCs do not define a specific approach for KMS to client communications, focusing instead on inter-device communications where interoperability is essential.

Cross Domain Working

So far we have only considered users communicating within a single domain using a single domain manager. To begin with, it is expected that this will be a common model; there will secure islands of communication for separate user groups.

As the service grows there will be a requirement to connect up these islands, both with each other and with other services. Initially this will be done via gateways, as specified in [5], but in the long term users using a client supporting MIKEY-SAKKE will be able to connect directly with true end-to-end security.

Shared information

As has already been stated, to communicate with users in a particular domain, the domain’s public certificate needs to be provided to the user’s client. It is anticipated that this will be provided by the user’s own domain manager.

For example, say the ‘dep.gov’ domain wishes to allow its users to call the users of the ‘nhs.uk’ domain. The ‘dep.gov’ domain manager obtains the NHS domain certificate from the NHS domain manager. This is provisioned (perhaps as signed XML) to the ‘dep.gov’ domain users, along with policies around communicating with the ‘nhs.uk’ domain.

Maintain your control

A central principle of MIKEY-SAKKE is to place the domain manager in control of user connections. By sharing its public certificate and by provisioning other domain’s public certificates to clients in its domain, the domain manager is able to define which connections are possible and which are not.

Certificate hierarchies have been intentionally avoided, as they transfer the question of ‘who do I trust?’ to remote authorities which may not know, and certainly do not own the risk implied by the question.

---

2 This assumes the user has been pre-provisioned with the address and PKI certificate of the KMS, and has login credentials.
Sector-Specific Services

Up to this point we have considered general services required by the majority of users. In this section we consider how sector-specific services can be simplified where MIKEY-SAKKE is already in use to provide communication security.

Compliance for Financial Firms

In banking, some areas of government, and a number of other industries, there are regulatory requirements that require communications between employees to be recorded so that the enterprise can support any future investigation into misconduct. At the same time, the enterprise rightly wishes to protect their business and control access to the content of these conversations.

MIKEY-SAKKE supports this use-case naturally by allowing an enterprise, such as a bank, to run its own domain manager and associated KMS. This allows the following benefits:

- Every communication over the network is secured.
- Storage of secured communications is possible without additional protections.
- It is not necessary to trust or manage the actual communication transport network. Only the KMS needs to reside within the bank’s control.
- The bank may provide time-limited access to any user’s communications when necessary by extracting appropriate keys from the KMS.

Secure Proximity Communications

Across a number of sectors there is a requirement to enable direct peer-to-peer communications between devices.

As MIKEY-SAKKE allows secure real-time communications without needing a current connection to network infrastructure, it is ideally suited to enable the security of communications over this interface.

Public-Safety Groups

To enable ‘push-to-talk’ group communications for public-safety users, a mechanism is required to distribute group keys to support these communications.

A commonly used mechanism is for a central group manager to be authorised to distribute keys. This is effective in the majority of contexts, but is not always sufficiently flexible. Determining which group manager is able to manage which groups is difficult and when the group manager cannot be reached, groups cannot be created or modified.

Using MIKEY-SAKKE as part of the group key distribution protocol solves these problems. Group managers can be keyed with the group identity, e.g. ‘usergroup@domain.com?uidyear=2014&uidmonth=03&uriusage=group’.

This allows them to sign group keys in a way that can be verified by group members, and distribute them encrypted to the group member’s identity. To take this a stage further, group managers can even be an end user, able to securely create and key groups on the fly without requiring a connection to network infrastructure.

The use of MIKEY-SAKKE in this context is currently under discussion in 3GPP.

Future Innovation

MIKEY-SAKKE has been designed to solve a specific problem: securing multimedia services. However, as an identity-based encryption scheme its uses may be far wider.

One company has used MIKEY-SAKKE to develop an encrypted webmail/web-storage service.

Another has demonstrated that it may be used to speed up TLS. In tests, MIKEY-SAKKE sped up the setup of HTTPS sessions by 7 times over standard TLS [15], meaning websites loaded over 200ms faster. This is due to the protocol only requiring only a single transmission from the client to the server. The use of SAKKE in this context may also remove the need for the server or client to maintain the ‘state’ necessary to support TLS requests.

It is also possible that there is significant benefit in using MIKEY-SAKKE to provide data encryption for cloud computing. This could remove the need for security assurance of distributed, virtualized infrastructure.

“MIKEY-SAKKE sped up the setup of HTTPS sessions by 7 times over standard TLS, meaning websites loaded over 200ms faster.”
Conclusion

There is an undeniable widespread need in government and industry for secure multimedia communications. To meet the need requires a new approach, as for twenty years, various forms of public key infrastructure have attempted to provide and failed to provide usable and scalable, client-to-client security.

MIKEY-SAKKE is an effective protocol for building a wide range of secure multimedia services for government and enterprise. As security is setup using only a single message, the protocol is highly scalable, flexible, and maintains availability as no domain infrastructure is required to communicate. In fact, access to domain infrastructure is only required on an occasional, such as monthly, basis.

Even though overhead on the domain infrastructure is minimal, with MIKEY-SAKKE the domain manager is in total control of the security of the domain, down to the cryptographic level. This is essential, as it allows the enterprise which is responsible for its risk and local regulations to be in control without that enterprise requiring any access or knowledge of the communication transport network. This means that secure communications may be provisioned just as simply as user devices are remotely managed by the enterprise today.

MIKEY-SAKKE uses well-established cryptographic techniques which have been subject to considerable academic scrutiny for over a decade. It is a publically available standard and there are no known licensing requirements or restrictions.

References


