Featuring Immediate Revocation in Mikey-sakke (FIRM)

Paulo Martins and Leonel Sousa
INESC-ID, IST
Universidade de Lisboa
Lisboa, Portugal
paulo.sergio@netcabo.pt, las@inesc-id.pt

Parashuram Chawan
SRUK
Samsung
Staines-upon-Thames, England
parashuram.chawan@samsung.com

***Abstract*—The use of Voice over Internet Protocol (VoIP) is becoming ubiquitous due to the multiple shortcomings of traditional Public Switched Telephone Network (PSTN) systems. As a result, the development of secure key establishment protocols is becoming increasingly important. The Communications-Electronics Security Group (CESG), in response to this demand, has published new key agreement protocols for the Multimedia Internet KEYing (MIKEY) protocol to provide low-cost secure VoIP communications, supported on Identity-based Public-Key Cryptography (IDPKC). In the context of IDPKC, the identity of users is used to derive their public-keys, which eliminates the expenses of maintaining a Public-Key Infrastructure (PKI). However, IDPKC systems suffer from inefficient user revocation and key renewal. In this paper, we take advantage of the fact that users need to be connected to the Internet to communicate, for introducing a SEcurity Mediator (SEM), who possesses a share of the users' private-keys, and with whom the users must cooperate, to sign and decrypt cryptograms. By taking advantage of this sharing, we introduce mechanisms to provide immediate user revocation and key renewal.**

***Keywords*-Key agreement; MIKEY-SAKKE; Secure Communication; VoIP;**

I. Introduction

The Multimedia Internet KEYing (MIKEY) protocol [1] defines a framework for key distribution. It provides a method for key exchange and source authentication designed for use in IP (Internet Protocol) multimedia scenarios. In particular, it may be used in conjunction with the Session Initiation Protocol (SIP) and the Secure Real-time Transport Protocol (SRTP) to provide secure Voice over Internet Protocol (VoIP) communication. Nevertheless, it has a wider range of applications.

The original protocol specified key distribution methods using pre-shared keys, the Rivest-Shamir-Adleman (RSA) cryptosystem, and, optionally, a Diffie-Hellman Key Exchange. With the intent of improving the latency of key distribution, since it is critical for real-time applications such as VoIP, while providing strong security, methods have been proposed to extend MIKEY. In particular, the Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY-SAKKE) [2] designed for use in IP Multimedia Subsystem (IMS) is herein focused on.

The MIKEY-SAKKE consists of an Identity-based Public-Key Cryptography (IDPKC) scheme based on Sakai and Kasahara (SAKKE) [3], and on Elliptic Curve-Based Certificaless Signatures for Identity-Based Encryption (ECCSI) [4]. In the context of IDPKC, the identity of users is used to derive their public-keys. A Key Management Service (KMS) is used as a root of trust and provisions the corresponding secret-keys to the users. Whereas SAKKE is used to wrap symmetric key-material, providing confidentiality, ECCSI is used for source authentication.

One of the most appealing features of IDPKC is that the public-keys are directly derived from the users' identifiers, removing the need for Public-Key or Certificate servers. However, in traditional IDPKC systems there is a need to periodically force users to change their identifiers, so that if a device is compromised, its public-key is eventually revoked from the system, by instructing the KMS not to provision the device anymore. In order to achieve this, usually the identifiers are concatenated with a time-stamp whose granularity dictates the validity period of a key. For instance, should Alice's identifier be *AliceOct2015*, it would only be valid during the month of October, 2015, but invalid from thereon. The main problem of this approach is that if the private-key of a user is compromised, an attacker may not only decipher messages directed to him but may also sign message in his behalf while the key is valid. It may therefore not be an acceptable solution for certain scenarios, e.g. corporate or government systems, where the credentials of a user should be immediately revoked upon compromise or removal of the user from the system.

Traditional IDPKC systems have another drawback. In the case of device being lost or stolen, but the user should remain in the system, there is no efficient way to renew his keys immediately. Concretely, since other users expect a certain Identifier, which is only updated on certain dates, and the private-key is tied to that Identifier, the immediate key renewal would comprise transmitting the new Identifier to all the users in the system.

The main contribution of this paper is a solution for the problem referred above of immediate revocation and key renewal, that we designate Featuring Immediate Revocation in Mikey-sakke (FIRM). We propose the introduction of

mediated computation of SAKKE decryption, and changing the underlying authentication mechanism to mediated IDPKC Elliptic Curve (EC) Schnorr signatures. By using SEcurity Mediators (SEMs), it is possible to implement efficient immediate revocation by instructing them to stop supporting the revoked users. Further, since private-keys are shared between the user and the SEM, even if the private-key share of the user is compromised, it is possible to produce a new user share, and accordingly update the SEM share, to effectively implement key renewal without changing the public Identifier.

The remainder of the paper is organized as follows. In Section II, we motivate the novel protocol by presenting one of many possible applications. In Section III, we present the related work of mediated IDPKC. An high-level design of the protocol is provided in Section IV, and the following two Sections discuss it in detail. Section VII explains the integration into MIKEY, and afterward, the protocol security is analyzed in Section VIII. Finally, conclusions are drawn.

II. Use Case

Enterprises as well as Government organizations are increasingly wishing to secure their communications. Traditional solutions based on the Public Switched Telephone Network (PSTN) suffer from multiple security issues, such as call diverting, rerouting, and evesdropping.

In contrast, VoIP leverages the Internet as an infrastructure for voice communication. For example, in the United Kingdom (UK), the Communications-Electronics Security Group (CESG) provisions standards and guidance to the Government and the Industry to ensure that appropriately assured products and services are available in the domain of Cyber Security. Recently, CESG has developed new key agreement standards, published as Internet Task Engineering Task Force (IETF) Requests for Comments (RFCs) to provide low-cost secure VoIP communications, which comprise the previously referred MIKEY-SAKKE protocol.

In this paper, we enhance the security features of MIKEY-SAKKE, by providing instant user revocation and key renewal. The typical use case consists of a system wherein users are able to hold sensitive voice or multimedia communications in disparate locations between two or more parties connected via untrusted networks, and where reliable mutual authentication is imperative. With the presented protocol, it is possible to immediately revoke users in the event of them being removed from the system. As an example, this may be triggered by an employee being made redundant or retired, or when his security clearance is removed. It is further possible to renew a user's key should it be compromised, without changing his identifier. This may be triggered, for instance, in the event of a device being stolen or lost, or an attacker having access to the user's key. Additionally, the KMS provides a way for lawful call intercept and monitoring.

III. Related Work

Boneh, Ding, Tsudik and Wong were one of the first to design a mediated cryptosystem [5]. The system is supported on a variant of RSA, in which the SEM is a semi-trusted server with whom the users must cooperate so as to decrypt and sign messages. Subsequently, Ding and Tsudik proposed an identity-based version of the mediated cryptosystem [6]. However, the scheme presents low security, since collusion between a user and the SEM leads to a total breakage of the system. This happens because the public modulus is shared among all the users and therefore complete knowledge of a private/public-key pair allows one to factor the modulus.

In order to avoid the aforementioned problem, mediated IDPKC systems supported on the Boneh-Franklin (BF) cryptosystem [7] have been designed, namely those in [8], [9]. We note that encryption using the BF scheme is significantly more expensive than with SAKKE, since it requires the computation of bilinear pairings. Nevertheless, as far as we know until now there were no mediated SAKKE schemes.

The ECCSI algorithm is based on Elliptic Curve Digital Signature Algorithm (ECDSA). There have been proposals on [10], [11] for the mediated computation of ECDSA signatures, which could be adapted to ECCSI. However, both schemes present drawbacks. The first requires a trusted center to pre-compute ephemeral secret-key shares for each signature, and the second requires the use of homomorphic encryption and zero-knowledge proofs techniques, both of which are computationally expensive.

Schnorr signatures provide a more efficient platform to introduce a SEM than ECDSA [12]. The original scheme is supported on Finite Fields, but it can be ported to Elliptic Curves (ECs). There have been proposals supported on Schnorr signatures to provide IDPKC without mediation [13], and mediation without IDPKC [14]. We are not aware of any proposal that combines the two. Other signature schemes supporting IDPKC and mediation are [15], [16]. The schemes are computationally demanding, since they are based on bilinear pairings, whereas EC Schnorr signatures are supported on EC point multiplication.

Further, there have been no proposals to introduce mediated cryptography in the context of the MIKEY-SAKKE protocol.

IV. High-level Design

In Figure 1, there is a high-level representation of the proposed protocol. The scheme depicts key provisioning, whereby a trusted entity, herein designated KMS, provides the SEM and the users with shares of the users' private-keys. We note that after this provisioning the users and SEMs can efficiently generate EC Schnorr signatures and decrypt cipher-texts together, but neither can alone.

It also depicts how a user, Alice, establishes a symmetric-key with another user, Bob. Parameters in red in the Figure are private, and parameters in black are public. Alice starts

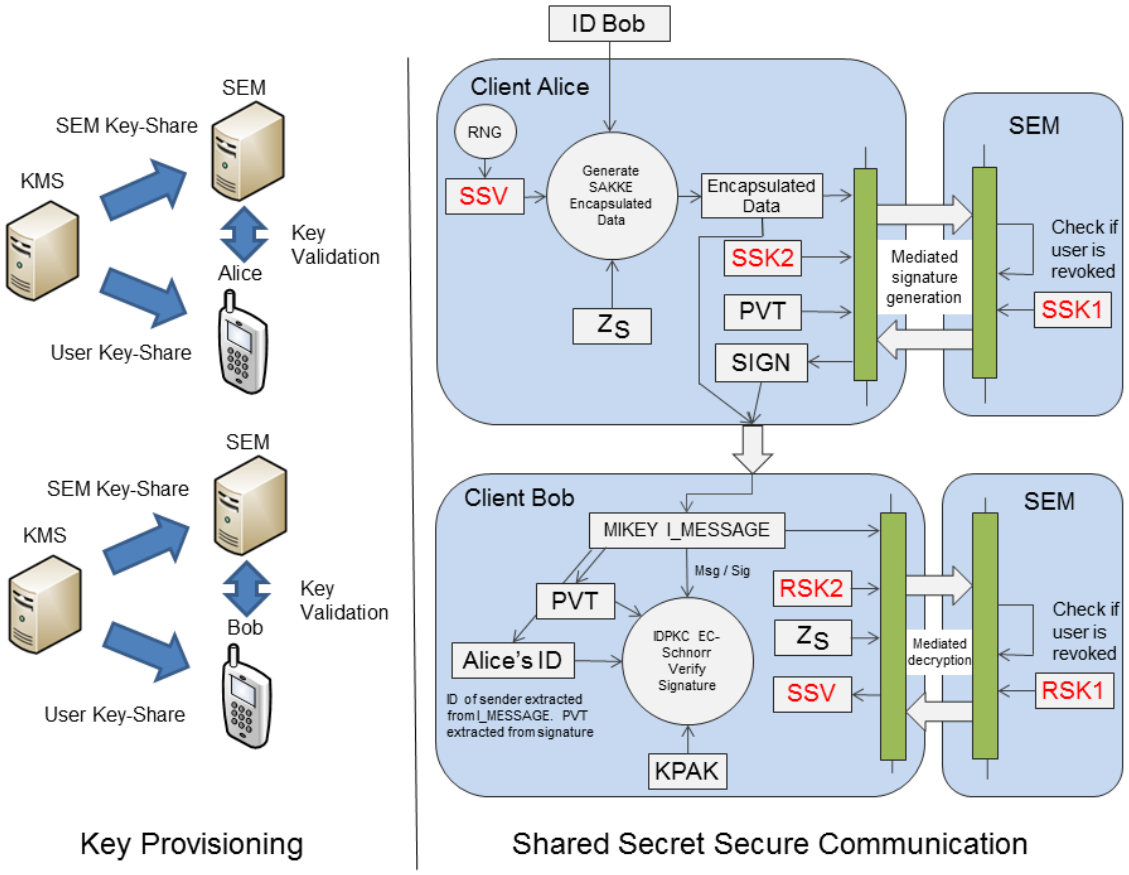

Figure 1. High-level Design of the mediated MIKEY-SAKKE protocol

by randomly generating a Shared Secret Value (SSV), which shall be later used to generate the symmetric-key material underpinning VoIP. Afterward, Alice encrypts the SSV by creating an Encapsulated Data as described in [17], using Bob's public-key. The Encapsulated Data is used to create an I_MESSAGE, as described in the MIKEY protocol [1], which is signed with help of the SEM. The SEM checks if Alice is revoked, and refuses to provide the service in that case. The I_MESSAGE is sent to Bob, who can verify the signature using EC Schnorr signature validation. If the signature is valid, Bob decrypts the SSV, by following the protocol described herein. The decryption protocol also requires interacting with the SEM, which checks if Bob is a valid identity, and refuses to provide the necessary service if that is not the case.

Unlike traditional IDPKC systems, it is possible to immediately revoke a user from the sytem by requesting the SEM to stop supporting that user. Further, even if the private-key share of a user is compromised, it is possible to create a new random share, and accordingly update the SEM's share, to effectively perform key renewal without changing the user's identifier. We describe in Sections V and VI, respectively, the proposed mediated encryption and signature protocols.

V. Mediate SAKKE scheme

The SAKKE cryptosystem is supported on a supersingular EC with equation $y^2 = x^3 - 3x \bmod p$, for a prime $p = 3 \bmod 4$, and on the Tate-Lichtenbaum pairing, herein denoted as $\langle , \rangle$. This pairing is a bilinear map that here is instantiated from $E(F_p)[q] \times E(F_p)[q]$ onto a subgroup of order q in PF_p, where $E(F_p)$ denotes an EC defined over the Finite Field F_p of order p; the notation $G[q]$ corresponds to the q-torsion of group G, and PF_p is the projectivization of F_p. PF_p is defined to be $(F_p^2)^*/(F_p)^*$, where F_p^2 is the extension field of degree 2 of the field F_p; herein, this field is instantiated as $F_p^2 = F_p[i]$, where $i^2 + 1 = 0$.

Elements of F_p are represented using integers from 0 to $p-1$. Elements of $F_p^2 = F_p[i]$ are represented as $x_1 + i \times x_2$, where x_1 and x_2 are elements of F_p. Elements of PF_p are cosets of $(F_p)^*$ in $(F_p^2)^*$ and can be written unambiguously in the form $x_1 + i \times x_2$. Descriptions of the Tate-Lichtenbaum pairing and of an $HashToIntegerRange(s, n, hashfn)$ function, which takes as input a string (s), a hash function

($hashfn$), and an integer (n), and returns a value between 0 and $n-1$, can be found in [17]. Moreover, we use the notation $a \leftarrow A$ to denote that a is uniformly and randomly selected from the set A.

We note that the Tate-Lichtenbaum pairing satisfies the following properties:

1) Bilinearity: $\langle [a]P, [b]Q\rangle = \langle P, Q\rangle^{a\times b}$, for any a, b in Z_q and P, Q in $E(F_p)[q]$;
2) Non-degeneracy: there exists P, Q in $E(F_p)[q]$ such that $\langle P, Q\rangle \neq 0$;
3) Computability: for all P, Q in $E(F_p)[q]$, $\langle P, Q\rangle$ can be efficiently computed.

The users of the system must agree beforehand on the public parameters of the underlying SAKKE encryption scheme, namely: i) an integer n, corresponding to the size of the symmetric-keys in bits exchanged by SAKKE; ii) a prime p, an odd prime q that divides $p+1$; iii) a point $P = (P_x, P_y)$ of $E(F_p)$ that generates the cyclic subgroup of order q, the value of $g = \langle P, P\rangle$; iv) and a cryptographic hash function, $Hash$. Additionally, the identities of the users are interpreted as integers.

A. Protocol

We start by describing how the KMS of a domain T, KMS_T, generates a master secret and the corresponding public-key. The KMS_T generates a master secret z_T, corresponding to a random integer in $\{2, \ldots q-1\}$, that is kept private. z_T is the foundation for the security of the system. The corresponding public-key is $Z_T = [z_T]P$, and is provisioned by the KMS_T to all who are authorized to send messages to the users of the IDPKC community.

We now describe the private-key share generation and validation process, depicted on the lefthand side of Figure 1. Upon the reception of a provisioning request from an user Alice, with identity a, the KMS randomly chooses a value z_1 from Z_q, and computes $z_2 = (a+z_T)^{-1} - z_1 \bmod q$, where the inversion is taken over the finite field F_q. Afterward, the KMS computes $RSK1 = [z_1]P$ and $RSK2 = [z_2]P$, and securely sends these values to the SEM and Alice, respectively, which correspond to part of the key-shares in the Figure. We note that $RSK1 + RSK2 = [(a+z_T)^{-1}]P$. Upon the reception of the private-key shares, the user and the SEM must validate them. In order to do that, Alice starts by checking that $RSK2$ lies in the EC and computing $w_2 = \langle [a]P + Z_T, RSK2\rangle$, and sends w_2, along with her identity, to the SEM. The SEM then checks if Alice's identity has been revoked from the system. If it has been, the SEM should send an error message to Alice, terminating the exchange. Otherwise, the SEM checks if RSK1 lies in the EC, computes $w_1 = \langle [a]P + Z_T, RSK1\rangle$, and sends this value to Alice. Finally, both Alice and the SEM check if $w_1 \times w_2 = g$.

The encryption operation takes place as described in Section 6.2.1 of [17], and is represented in Figure 1 as a circle labeled "Generate SAKKE Encapsulated Data". We suppose that Alice, provisioned by KMS_T, wants to encrypt and send an SSV to Bob, who has been provisioned by KMS_S. Encryption operates as follows: Alice randomly generates an SSV, and computes $r = HashToIntegerRange(SSV||b, q, Hash)$. Afterward, Alice multiplies Bob's public-key $[b]P + Z_S$ by r, obtaining $R_{b,S} = [r]([b]P + Z_S)$, and uses g^r to mask the value of SSV, by computing $H = SSV$ XOR $HashToIntegerRange(g^r, 2^n, Hash)$. The cipher-text corresponds to the tuple $(H, R_{b,S})$.

The decryption operation requires Bob to interact with the SEM, and is denoted "Mediated decryption" in the Figure. Bob starts by sending $R_{b,S}$ and his identity to the SEM. The SEM checks if the user has been revoked from the system, and replies with an error message if that is the case. If it is not the case, the SEM applies the bilinear pairing to $R_{b,S}$ and his share of the private-key, $RSK1$, to obtain w_1. The value w_1 is then transmitted to Bob. Bob symultaneously computes $w_2 = \langle R_{b,S}, RSK2\rangle$ and afterward multiplies it by w_1. We note that by bilinearity, $w = w_1 \times w_2 = \langle R_{b,S}, RSK1\rangle \times \langle R_{b,S}, RSK2\rangle = \langle R_{b,S}, RSK1 + RSK2\rangle = \langle [r(a+z_T)]P, [(a+z_T)^{-1}]P\rangle = \langle [r]P, P\rangle = g^r$. SSV is then reconstructed as $SSV = H$ XOR $HashToIntegerRange(w, 2^n, Hash)$. Then, $r = HashToIntegerRange(SSV||b, q, Hash)$ is calculated, and $R_{b,S}$ is reconstructed as $TEST$, from the value of r. If $TEST$ matches the value of the received $R_{b,S}$, SSV may be used to derive key material for the application to by keyed; otherwise the SSV must not be used.

VI. Mediated IDPKC EC Schnorr Signatures

The mediated IDPKC EC Schnorr signatures are supported on ECs with equation $y^2 = x^3 - 3x + B$ over F_p, for a prime number p of size n_1 bits, and B an element of Z_p. We assume the users have agreed beforehand on a particular EC, E, defined over some Finite Field F_p, and on a point G on the EC that generates a subgroup of order q, where q is a prime number of size n_2 bits. We further assume that they have agreed on a cryptographic hash function, denoted $Hash$, that maps arbitrary strings to strings of N_1 bytes, where $N_1 = Ceiling(n_1/8)$. We also define $N_2 = Ceiling(n_2/8)$.

A. Description of the Protocol

The KMS generates a master secret $KSAK$, corresponding to an integer randomly chosen in $\{2, \ldots, q-1\}$, that is kept private, and is the second root of trust for the system. The corresponding public-key $KPAK$ is computed as $KPAK = [KSAK]G$ and must be provisioned to the users in a trusted fashion.

The IDPKC EC Schnorr Signatures private-key shares are generated upon the request of provisioning by a user, and are transmitted during the process depicted on the lefthand

side of Figure 1. When the KMS receives a request for generating the shares for an identity a, it randomly generates a value v in Z_q. The EC point $PVT = [v]G$ is afterward generated, and $HS = hash(G||KPAK||a||PVT)$ is computed. The first share $SSK1$ is randomly selected from Z_q, and $SSK2$ takes the value of $SSK2 = (v + KSAK \times HS) - SSK1 \bmod q$. It should be noted that $SSK = SSK1 + SSK2 = v + KSAK \times HS \bmod q$. The pairs $(SSK1, PVT)$ and $(SSK2, PVT)$ are sent to the SEM and Alice, respectively, as part of the key-shares of Figure 1.

After receiving the private-key shares, Alice and the SEM should check their validity. The process of validation consists of Alice computing $[SSK2]G$, and sending the resulting EC point, along with her identity to the SEM, the SEM checking if Alice's identity has been revoked, and replying with $[SSK1]G$ if it has not been. Afterward, both Alice and the SEM compute $HS = hash(G||KPAK||a||PVT)$, and $Y = [SSK1]G + [SSK2]G$, and check if the derived public-key, Y, matches the public-key computed as $PVT + [HS]KPAK$.

After Alice having validated her private-key, she can compute EC Schnorr signatures, as depicted in Figure 1 in the "Mediated signature generation" process. She starts by randomly choosing a value j_c in Z_q, and computing $J_c = [j_c]G$. The values of J_c, the message being signed, M, and the identity a, are transmitted to the SEM, which checks if Alice's identity has been revoked. If it has been revoked, the SEM should reply with an error message. Otherwise, it verifies that J_c lies in E, randomly selects j_s and computes $J_s = [j_s]G$. J_s and J_c are added to produce $J = J_s + J_c$. The value of HS, computed during the validation phase, is combined with the x-coordinate of J, $r = J_x$, and with the message, M, to produce $HE = hash(HS||r||M)$. Afterward, the SEM computes $s_s = j_s - SSK1 \times HE \bmod q$, and transmits this value, along with J_s, to Alice. Alice, after verifying that J_s lies in E, can now compute $J = J_c + J_s$, and $HE = hash(HS||r||M)$, where r is the x-coordinate of J. Then, Alice computes her share of s as $s_c = j_c - SSK2 \times HE \bmod q$. Afterward, she combines s_s and s_c to produce $s = s_c + s_s$, and outputs the triplet (HE, s, PVT) as the signature. We note that, unlike ECCSI, s requires N_2 bytes to be represented.

We now assume that Bob wants to validate a signature produced by Alice, in the process labeled as "IDPKC EC-Schnorr Verify Signature" in Figure 1. Bob, upon receiving Alice's signature triplet (HE, s, PVT), computes her public-key, by first producing the hash $HS = hash(G||KPAK||ID||PVT)$ and then calculating $Y = PVT + [HS]KPAK$. Afterward, he extracts HE and s, and produces $J = [s]G + [HE]Y$. Using HS, the x-coordinate of J and the message M, Bob reconstructs the value HE as HE'. Then, Bob accepts the signature if $HE = HE'$, and otherwise rejects it.

VII. Integration with MIKEY

The described methods for IDPKC encryption, decryption, signature generation and signature validation should be used within the MIKEY protocol. MIKEY is a key management protocol that is intended for use within real-time applications, and can be used to set up encryption keys for multimedia sessions that are secured using SRTP. The proposed MIKEY mode, herein designated FIRM, requires a single simplex transmission, where the Initiator sends a message containing the SSK encrypted as SAKKE Encapsulated Data and a signature to the intended recipient. The Responder then validates the signature, and processes the Encapsulated Data to obtain the SSV. The SSV is then used to derive symmetric key-material for the keying application, namely VoIP. As a result of these steps, both parties are mutually authenticated.

VIII. Security Considerations

We note that in [18] the SAKKE scheme is proven to be semantically secure against adaptive chosen cipher-text attacks, under the random oracle model, assuming the hardness of the q-Bilinear Diffie-Hellman Inverse problem. Further, the mediated version of the scheme can be proven to be weakly semantically secure against adaptive chosen cipher-text attacks by adapting the proof of security for mediated BF encryption scheme in [8] to SAKKE. This means that no coalition of dishonest users with the SEM can allow them to find any information about a cipher-text intended to an honest user.

Additionally, the original IDPKC Schnorr signature scheme (without mediation) is proven in [13] to be secure against existential forgery on adaptively chosen message and identity attacks, under the random oracle model, assuming the hardness of discrete logarithms on the underlying subgroup of the EC. We also note that the security proof for the 2-message mediated Schnorr signatures of [14], which states that the mediated Schnorr signature scheme is secure against malicious clients – assuming the hardness of discrete logarithms – and secure against malicious servers – assuming the hardness of the known-target discrete logarithm problem – can be adapted to the EC IDPKC Schnorr signature scheme, by changing the underlying group to the EC.

In order to tolerate multiple compromises of each party, it is recommended to periodically update the shares of the private-key. Clearly, an adversary that compromises both parties between two successive key updates will learn the complete state of the system and break the security of the public-key; however, the system should be able to withstand multiple break-ins as long as there is one honest key update in between. This can be achieved by having Alice computing a random integer d, in the range from 1 to $q-1$ and securely sending it to the SEM. The SEM and Alice may then update their key shares as $RSK1' = RSK1 + [d]P$ and $RSK2' = RSK2 - [d]P$, respectively. Alice should also compute another random value d' in Z_q and securely transmit it to the

SEM. Both the SEM and the user may then update their key shares $SSK1$ and $SSK2$ to $SSK1' = SSK1 + d' \bmod q$ and $SSK2' = SSK2 - d \bmod q$. These exchanges should be protected using a secret that has not been compromised. Alternatively, if Alice's private-key shares have been stolen (i.e. she no longer has access to it, but someone else has), she can request the KMS for a new provisioning, effectively renewing her private-key share without the need to update her Identifier.

IX. Conclusion

The use of IDPKC has been recently proposed by CESG, in the form of the MIKEY-SAKKE protocol, as an alternative to traditional certificate-based public-key encryption solutions, for key management in multimedia solutions. Whereas IDPKC systems eliminate the need for PKIs, since public-keys are derived from users' identifiers, they suffer from inefficient user revocation and key renewal. The proposed protocol, FIRM, differs from current technology by introducing mediated encryption within MIKEY-SAKKE, and achieving efficient immediate revocation of users and key renewal as a byproduct. In order to achieve this, new elements had to be produced, namely a mediated version of the SAKKE scheme, and a mediated version of IDPKC EC Schnorr signatures.

Whereas most traditional IDPKC systems force users to renew their keys monthly or weekly to reduce the impact of compromised keys, with FIRM this need is eliminated, due to the immediate revocation feature. We further note that compromised users are not able to decipher neither old nor recent messages, since the SEM will not cooperate with them. This is in contrast to Certificate-based encryption schemes, where even after a user's Certificate has been revoked, he can still decipher old messages, and is able to decipher messages transmitted by users who have not been informed of his revocation.

Acknowledgment

The authors would like to thank HiPEAC Industrial PhD Internships, for creating this opportunity for the collaboration between INESC-ID and SRUK.

References

[1] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and Norrman, K., "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[2] Groves, M., "MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY)", RFC 6509, February 2012.

[3] Sakai, R., Ohgishi, K., and Kasahara, M., "ID based cryptosystem based on pairing on elliptic curves", Symposium on Cryptography and Information Security - SCIS, 2001.

[4] Groves, M., "Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)", RFC 6507, February 2012.

[5] Boneh, D., Ding, X., Tsudik, G., and Wong, C., "A method for fast revocation of public key certificates and security capabilities", in Proceedings of the 10th conference on USENIX Security Symposium - Volume 10, ser. SSYM'01. Berkeley, CA, USA: USENIX Association, pp. 22-22, 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251327.1251349

[6] Ding, X., and Tsudik, G., "Simple identity-based cryptography with mediated RSA", in RSA, Proceedings CT-RSA 2003, LNCS 2612, Springer-Verlag, 2003.

[7] Boneh, D., and Franklin, M., "Systems and methods for identity-based encryption and related cryptographic techniques", US patent 7,113,594, 2006.

[8] Libert, B., and Quisquater, J., "Efficient Revocation and Threshold Pairing Based Cryptosystems", Principles of Distributed Computing (PODC) 2003.

[9] Chow, S., Boyd, C., and Nieto, J., "Security-Mediated certificateless cryptography", in Proceedings of the 9th international conference on Theory and Practice of Public-Key Cryptography (PKC'06), Yung, M., Dodis, Y., Kiayias, A., and Malkin, T. (Eds.), Springer-Verlag, Berlin, Heidelberg, pp. 508-524, 2006. [Online]. Available: http://dx.doi.org/10.1007/11745853_33

[10] Langford, S., "Threshold DSS Signatures Without a Trusted Party", CRYPTO '95 (LNCS 963), pp. 397-409, 1995.

[11] MacKenzie, P., and Reiter, M., "Methods and apparatus for two-party generation of DSA signatures", US patent 20,030,059,041, 2003.

[12] Schnorr, C., "Method for identifying subscribers and for generating and verifying electronic signatures in a data exchange system", US patent 4,995,082, 1991.

[13] Chatterjee, S., Kamath, C., and Kumar, V., "Galindo-Garcia identity based signature revisited", in Information Security and Cryptology - ICISC 2012, volume 7839 of Lecture Notes in Computer Science, Kwon, T., Lee, M., and Kwon, D. (Eds.), Springer Berlin / Heidelberg, pp. 456-471, 2013. [Online]. Available: http://eprint.iacr.org/2012/646

[14] Nicolosi, A., Krohn, M., Dodis, Y., and Mazieres, D., "Proactive Two-Party Signatures for User Authentication", NDSS, 2003.

[15] Cheng, X., Guo, L., and Wang X., "An Identity-based Mediated Signature Scheme from Bilinear Pairing", International Journal of Network Security, Vol.2, No.1, pp.29-33, 2006.

[16] Wang, X., and Wang, S., "An Identity-Based Mediated Signature Scheme Without Trusted PKG", Informatica 35, pp. 83-90, 2011.

[17] Groves, M., "Sakai-Kasahara Key Encryption (SAKKE)", RFC 6508, February 2012.

[18] Chen, L., Cheng, Z., Malone-Lee, J., and Smart, N., "Efficient ID KEM based on the Sakai-Kasahara key construction", IEEE Proceedings of Information Security, 2006.