



**HAL**  
open science

# A Trustful Authentication and Key Exchange Scheme (TAKES) for ad hoc networks

Tony Cheneau, Andrei Vlad Sambra, Maryline Laurent

► **To cite this version:**

Tony Cheneau, Andrei Vlad Sambra, Maryline Laurent. A Trustful Authentication and Key Exchange Scheme (TAKES) for ad hoc networks. NSS 2011: 5th International Conference on Network and System Security, Sep 2011, Milan, Italy. pp.249–253, 10.1109/ICNSS.2011.6060009 . hal-00745353

**HAL Id: hal-00745353**

**<https://hal.science/hal-00745353>**

Submitted on 25 Oct 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Trustful Authentication and Key Exchange Scheme (TAKES) for Ad Hoc Networks

Tony Cheneau  
National Institute of Standards and Technology  
Gaithersburg, MD, USA.  
Email: tony.cheneau@nist.gov

Andrei Vlad Samba, Maryline Laurent  
TELECOM SudParis  
CNRS Samovar UMR 5157  
9 rue Charles Fourier, 91011 Evry, France.  
Email: andrei.samba@it-sudparis.eu  
Email: maryline.laurent@it-sudparis.eu

**Abstract**—This paper presents a new public key distribution scheme adapted to ad hoc networks called TAKES for Trustful Authentication and Key Exchange Scheme. Its originality lies in performing authentication and key distribution with no need for a trusted authority or access to any infrastructure-based network, thanks to the use of Cryptographically Generated Addresses. Moreover the solution is very convenient having a simple operational mode at no extra hardware cost.

TAKES aims to build a trust association between a person, his/her communicating device, the IP address of the device, and his/her public key. As a direct result, new security functions like associating a misbehaving node to its owner, securing end-to-end communications through tunnels, or even implementing a light naming system can be enabled on top of ad hoc networks. TAKES is formally proven using BAN logic and a proof-of-concept implementation demonstrates its feasibility within ad hoc networks.

**Keywords**—ad hoc network, authentication protocol, public key distribution scheme, cryptographically generated addresses, passphrase authentication

## I. INTRODUCTION

Security remains one of the most challenging issues in ad hoc networks. So far, most of the secure routing proposals like ARAN, SEAD, Ariadne, SPINS, and SRP ([1], [2], [3], [4], [5]) assume that all the (honest) principals are sharing a secret and/or public keys. Other approaches like distributed certification authorities [6] or threshold cryptography schemes [7] assume unrealistically that devices have enough computational resources. To cope with these strong assumptions, new mechanisms must be introduced, accommodating the trust scenarios specific to ad hoc networks.

In this paper, we define a security mechanism adapted to ad hoc networks, called TAKES for Trustful Authentication and Key Exchange Scheme. It should be noted that TAKES can equally operate over any type of TCP/IP network topology. TAKES enables two or more people to securely distribute their public key, and enables applications to take advantage of the keys for Virtual Private Network (VPN) establishment, securing routing protocols... Participants must be physically close to their own communicating device (e.g. notebook, PDA, smartphone) before activating an Out-of-Band Channel (OOBC), like voice or sign language, and distributing their public keys. Additionally, participants are not assumed to

implicitly trust devices and/or their administrators on the network. Finally, there are no assumptions on the availability of some network infrastructures (e.g. access points, routers, switches, gateways, etc.).

TAKES relies on Cryptographically Generated Addresses (CGA) [8] for securing broadcasted messages. CGAs are specific IPv6 addresses that cryptographically bind a public key to an address. CGAs bring advantages to TAKES as it makes possible building an association between the address, its owner's public key and device, and as such, new security applications into ad hoc networks are enabled.

This paper is structured as follows. Section II presents the related works about the public key exchange performed through OOBC, with their weaknesses and limitations. Section III provides an overview of our proposal. Sections IV and V respectively detail the messages used for key distribution and for key update/revocation. Section VI is dedicated to formally proving our solution by using the BAN logic. Section VII analyzes the security aspects of TAKES. In Section VIII, a proof-of-concept implementation, tested in an ad hoc network environment, demonstrates the feasibility of TAKES. Finally, Section IX gives conclusions and future work perspectives.

## II. RELATED WORKS

As a fundamental assumption in this article, we consider that there are no security infrastructures, no trusted third parties (TTPs) and no prior trust relationships between nodes in an ad hoc network. Under these specific constraints, security features can only be provided by soliciting the users operating the nodes. However, it is common knowledge that users are not good at remembering long strings and performing arithmetic computations. On the other hand, users are better at performing computationally harder tasks like pattern recognitions, or physical interactions such as placing two objects in front of each other. In [9], Perrig and Song propose a solution named *Random Arts* where public keys are transferred through an OOBC, under the form of a digital image which has to be evaluated by users. Similarly, E. Kim et al. [10] suggest that a human user compares a short text message to share an identical temporal key between two devices.

OoBC are generally employed with pairing schemes to establish a secret between two participants. Feeny et al. [11] use infrared communications to transmit cryptographic material (e.g. public key) between two nodes willing to establish a trust relationship. Still based on the infrared channel, Balfanz et al. [12] propose a mechanism that carries only critical information through the channel. A non-secured channel is then used to determine the authenticity of the entities and to speed up the public key exchange. Numerous contributions have since been proposed. Among them, we note an interesting proposal based on an OoBC composed of a LED matrix and a Webcam [13], and another proposal based on a LASER technology [14]. MacCune et al. [15] proposed the "Seeing-is-Believing" (SiB) method which uses two unidirectional visual OoBCs to display and read 2D barcodes.

Recently, Goodrich et al. [16] introduced the "Loud and Clear" (L&C) system, based on an audio OoBC along with vocalized MadLib sentences derived from the hash of a device's public key. This solution implies the use of a speaker on one device and a speaker or a display on the other one. The user is required to compare the two MadLib sentences which contain 8 words and make a decision whether to accept or abort the device pairing. Claycomb and Shin [17] propose a key establishment method for mobile devices, called UbiSound. Using an audio OoBC, two devices can securely transmit verification of the key establishment information between two mobile devices. Their solution eliminates the audio based human-verification components specific to most of the OoBC pairing methods.

In addition, Montenegro and Castelluccia describe an OoBC mechanism [18] based on the Statistically Unique and Cryptographically Verifiable (SUCV) identifiers. In this scheme, participants generate a SUCV, i.e. a crypto-based identifier, which is cryptographically binding the public key of the participant to an identifier. The specificity of this identifier is that proof of ownership can be established, so no identifier spoofing can be performed. The participants are asked to convert their identifiers into a sentence, where each word is extracted from a specific dictionary and represents a set of bits. When a participant intends to communicate his/her public key to another user, he/she reads the corresponding sentence (i.e. oral communication). The receiver can then convert the sentence into an identifier and retrieve the associated public key. The originality of this work lies in that no specific hardware is needed.

The previous solutions focus only on *authenticating* the connection, and do not perform *device authentication* (i.e. pairing in the presence of multiple potentially pairable devices). In our solution, we are interested to offer both authentication and identification of people and devices at the same time, thus creating a trust association between a person and his/her communicating device, the IP address of the device and the person's public key.

It should also be noted that compared to existing works, our solution does not require any specialized equipment ([13] and [14]) and has no line of sight constraints ([11] and [12]).

Moreover, it is only composed of simple actions. Furthermore, it enables the distribution of a public key not only to a single node but to a whole network [18], thus fitting conference-like scenarios (where people are considered to be physically close).

### III. OVERVIEW OF TAKES

TAKES supports multi-hop distribution of a public key bound to its owner's identity within an ad hoc network. Here, the term "participant" designates both the nodes distributing their public key through TAKES as well as the ones that are only listening. Participants willing to broadcast their public key are assumed to generate a key pair (e.g. RSA or ECC). They are then identified by their CGA addresses [8] which are addresses cryptographically linked to their own public key.

Introducing the CGA addresses is of high benefit for the participant which can make straight use of any CGA-based secure protocols like SEND [19] and CGA-IKE [20]. TAKES helps strengthening the security of these protocols in some specific scenarios, and thus could lead the participant to favor these protocols for securing its communications. These relevant scenarios are not detailed in this paper due to space constraints.

For distributing its public key, a participant is sending two TAKES messages. As will be discussed in Section VII, the order of the messages is of utmost importance in order to prevent Man-in-the-Middle attacks.

The first message, also referred to as "link message", is broadcasted to all the other participants through the ad hoc network. This message contains the public key of the participant and several public elements, such as the equipment name (e.g. *notebookA*). It is protected with a digital signature generated with the private key of the participant and a Hash-based Message Authentication Code (HMAC) keyed with a (one-time) secret passphrase.

The second message is broadcasted through an Out-of-Band Channel, such as voice, and it only contains the secret passphrase used to verify the HMAC contained in the link message. This message must be emitted by a publicly authenticable OoBC, such as voice, so that the public key can be directly linked to the participant (i.e. a human user). Due to the specificity of this channel (e.g. oral communication), OoBC messages might be lost (i.e. people not paying attention). To cope with possible losses of OoBC messages, the sender has to make sure that the participants are listening (i.e. by drawing their attention) and it might retransmit the message multiple times if necessary.

Upon receiving both messages, the participants can authenticate the first message by checking the HMAC of the first message against the passphrase contained in the OoBC message. If the message authenticity is successfully checked, each participant can bind the sender's identity (i.e. a human user), its public key, its equipment's name and its CGA address. This tuple is then stored by the participants so each piece of information can be retrieved for later use.

Note that only unidirectional communication is used (through two different channels) by the participant transmit-

ting its public key. Consequently, no (negative or positive) acknowledgement is performed or required from the receivers through the ad hoc network.

Let us give a short illustrative scenario example by considering a small conference room where a meeting between different departments takes place. Participants know and trust each other either implicitly (as colleagues), or explicitly (proving their identities). In order for everyone to distribute its public key, participants take turns in broadcasting a TAKES message. Participant “A” first draws attention to its intention to broadcast a TAKES message. Then, it can start broadcasting the message through the ad hoc network. If all users have successfully received the message, or if no user is reporting problems, participant “A” introduces itself and broadcasts its secret passphrase through the OOB: ‘Hello! My name is participant A” and my secret passphrase is “unique passphrase”.’

#### IV. KEY BROADCAST MESSAGES

The two-message TAKES protocol is depicted in Figure 1 and considers the four following steps:

- 1) The initiating node generates its CGA address  $@A$ . This address is the concatenation of the two following elements: a subnet prefix  $subnetA$  and the result of the application of the hash function SHA-1 over the subnet prefix  $subnetA$ ,  $userA$ ’s name ( $nameA$ ), the name of its equipment ( $equipA$ ), and the public key ( $pkA$ ). The procedure can be summed up by the following formula, where  $|$  is the concatenation function and the  $trunc64()$  is a truncation function that returns the 64 leftmost bits of the input string.

$$@A = [subnetA | trunc64(sha1(subnetA, nameA, equipA, pkA))]$$

More details on the CGA generation process can be found in [8]. It should be noted that this step can also be performed offline and hence it does not impair or delay the transmission of messages.

- 2) The link message sent over the (in-band) link channel. The participant  $userA$  distributes its public key ( $pkA$ ) to all TAKES participants within the network (i.e. nodes subscribers to the multicast group) by sending a multicast message, signed with its private key  $prA$ . The message contains the following elements: the address of  $userA$ ’s node ( $@A$ ) used as the source address of the message, the name of  $userA$  ( $nameA$ ), the name of its equipment ( $equipA$ ), a timestamp to prevent replay attacks ( $tsA$ ), its public key ( $pkA$ ) and a HMAC computed over  $pkA$  and  $tsA$  and keyed by a one-time secret ( $secretA$ ). The secret  $secretA$  is a passphrase that  $userA$  discloses to the other participants in the next step over the OOB.
- 3) The secret passphrase ( $secretA$ ) is transmitted over the OOB. The receivers are then prompted with an option to register the public key contained in the received message ( $pkA$ ). To do so, they are required to type in the passphrase ( $secretA$ ) and to know the name of user

A ( $nameA$ ), which are both communicated by the sender via the OOB. The OOB is an authenticable channel such as an oral communication (e.g. “Hello, my name is  $userA$  and my passphrase is  $secretA$ ”).

- 4) Each receiver verifies the authenticity of the message from the link channel. First, it verifies that  $secretA$  validates the HMAC, thus proving the link between  $userA$  and its public key ( $pkA$ ). Second, the participant verifies the freshness of the timestamp contained in the link channel message ( $tsA$ ). Third, the authenticity and integrity of the link message are confirmed through the verification of the digital signature  $sig_{prA}$ .

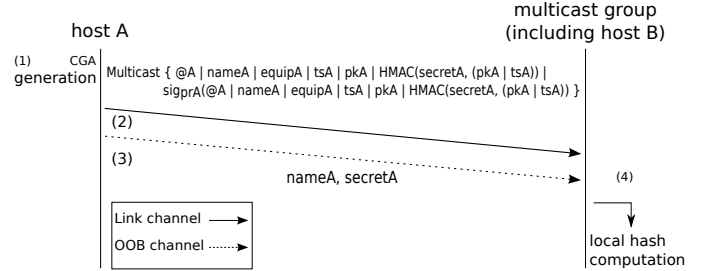


Fig. 1. TAKES key exchange.

After receiving and validating these messages, each participant obtains and stores the public key of the initiator. Furthermore, participants are able to link the initiating node (here, a person named  $userA$ ), its name ( $nameA$ ), its equipment’s name ( $equipA$ ), its IP address ( $@A$ ) and its public key ( $pkA$ ).

Note that for verifying the timestamp’s freshness, the synchronisation of the clock is required for all the participants in order to prevent replay attacks. This can be achieved in ad hoc networks by using an adapted clock synchronisation protocol [21]. However, in practice it is difficult to provide security for this kind of synchronisation protocol, so it is also possible to rely on a simpler mechanism such as the timestamp validation procedure. This procedure is described in the SEND protocol [19] and it allows nodes having loosely synchronized clocks to communicate.

#### V. KEY UPDATE AND REVOCATION MESSAGE

TAKES is complemented with a key update and revocation scheme. Note that the key revocation is a sub-case of key update as for updating a key, a key revocation is done. As such, we focus mainly on the key update scheme, highlighting, when necessary, the differences between them. The key update message format is illustrated in Figure 2. It does not include all the components of the initial authentication mechanism, as the identity of user A (i.e.  $@A$ ,  $nameA$ ,  $equipA$ ,  $pkA$ ) is known and a sufficient trust level has already been established. The IP address  $@A$  is used to lookup the identity of the sender. Authenticity of the message is ensured by signing the message with the previous secret key. It should be noted that this process does not fully guarantee key revocation (like the certificate revocation process in PKI), instead, it provides a mean to indicate that a key should no longer be used.

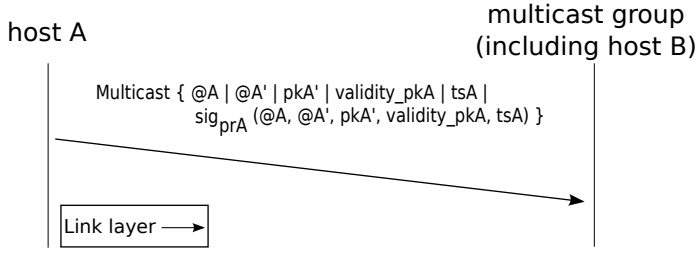


Fig. 2. Key update and revocation.

The message includes the following elements:

- 1) the IP address @A. This address enables the receivers to lookup the identity of the sender in their local database;
- 2) the current IP address @A'. This IP address is likely to be different from the previous one if the public key is updated, or it remains unchanged if the purpose of the message is only to revoke a key and not to perform updating;
- 3) a public key  $pkA'$ . In case of revocation only, the public key is the same key as  $pkA$ . In case of updating, the key  $pkA'$  is different and is meant to replace the previous key  $pkA$  after expiration of the date  $validity\_pkA$ ;
- 4) a validity date  $validity\_pkA$  indicates when the public key  $pkA$  is set to expire. That is, if the date is prior or equal to the current time, the old key is no longer valid. If the date is set to a date in the future, the old public key is set to expire, and can still be utilized for current connections, in parallel with the new key (if provided);
- 5) a timestamp value  $tsA$  helps preventing replay attacks;
- 6) the signature  $sig_{prA}$  ensures the authentication of the message. We still consider the public key  $pkA$  to be valid at the moment the message is received.

## VI. FORMAL PROOF

TAKES messages have been formally proven using the Burrow-Abadi-Needham (BAN) logic [22]. The BAN formalism is based on a many-sorted modal logic where several types of objects are distinguished: principals, encryption keys, and statements. The BAN logic has the advantage over most of the formal validation tools that it makes it possible modeling both the OOB channel and the trust between users in real life.

### A. Key Broadcast Messages

Using the BAN notation, TAKES messages can be represented as follows:

Link channel:  $A \longrightarrow B : \{Xa, Ta, \overset{Ka}{\mapsto} A, \langle \overset{Ka}{\mapsto} A, Ta \rangle_S\}_{Ka^{-1}}$

OOB channel:  $A \longrightarrow B : \{A \overset{S}{\rightleftharpoons} B\}_{K_{IDa}^{-1}}$

In these expressions,  $A$  and  $B$  are principals,  $Xa$  is comprised of the address of  $A$ , the name of  $A$  and the name of the

equipment  $A$ ,  $Ta$  is a timestamp generated by  $A$ ,  $Ka$  and  $Ka^{-1}$  are respectively the public and the private key of  $A$  and  $S$  is a one-time passphrase. We also model the OOB by introducing  $IDa$ , the identity of the principal  $A$ ;  $K_{IDa}$ , the public key associated to  $IDa$ ; and  $K_{IDa}^{-1}$ , the private key associated to  $IDa$ . While this public/private key pair does not really exist, it serves to model the authenticity of the OOB. Hence, when a participant is "speaking" using the OOB, they implicitly sign all their messages to prove their authenticity (e.g. in an oral communication the voice of the speaker and the lip synchronization proved the authenticity, thus confirming which person is speaking).

BAN logic assumptions are defined in Table I.

TABLE I  
INITIAL ASSUMPTIONS FOR KEY DISTRIBUTION.

1. $B \equiv \overset{K_{IDa}}{\mapsto} IDa$
2. $B \equiv A \overset{Ka}{\mapsto} A$
3. $B \equiv \#(Ta)$
4. $B \equiv \#(A \overset{S}{\rightleftharpoons} B)$
5. $B \equiv A \overset{S}{\mapsto} Xa$
6. $B \equiv IDa \overset{S}{\mapsto} B$

Table II describes the steps for the protocol verification. The link layer message is assumed to be already received and the verification starts by analyzing the message sent over OOB. Note that CGA aspects are not part of TAKES, so we do not consider them in BAN logic.

TABLE II  
PROTOCOL VERIFICATION STEPS.

$A \longrightarrow B : \{A \overset{S}{\rightleftharpoons} B\}_{K_{IDa}^{-1}}$	
7. $B \triangleleft \{A \overset{S}{\rightleftharpoons} B\}_{IDa^{-1}}$	
8. $B \equiv IDa \triangleleft A \overset{S}{\rightleftharpoons} B$	// (1), msg-meaning rules
9. $B \equiv IDa \equiv A \overset{S}{\rightleftharpoons} B$	// (4)
10. $B \equiv A \overset{S}{\rightleftharpoons} B$	// (6), jurisdiction rule
$A \longrightarrow B : \{Xa, Ta, \overset{Ka}{\mapsto} A, \langle \overset{Ka}{\mapsto} A, Ta \rangle_S\}_{Ka^{-1}}$	
11. $B \triangleleft \{Xa, \overset{Ka}{\mapsto} A, Ta, \langle \overset{Ka}{\mapsto} A, Ta \rangle_S\}_{Ka^{-1}}$	
12. $B \triangleleft \langle \overset{Ka}{\mapsto} A, Ta \rangle_S$	// (11)
13. $B \equiv A \triangleleft (\overset{Ka}{\mapsto} A, Ta)$	// (10), (12)
14. $B \equiv \#(\overset{Ka}{\mapsto} A, Ta)$	// (3), freshness rule
15. $B \equiv \overset{Ka}{\mapsto} A$	// (2), (13), (14)
16. $B \equiv A \triangleleft (Xa, \overset{Ka}{\mapsto} A, Ta, \langle \overset{Ka}{\mapsto} A, Ta \rangle_S) // (11), (15)$	
17. $B \equiv A \triangleleft (Xa, \overset{Ka}{\mapsto} A, Ta)$	// (16), once-said rule
18. $B \equiv \#(Xa, \overset{Ka}{\mapsto} A, Ta)$	// (3), (17)
19. $B \equiv A \equiv (Xa, \overset{Ka}{\mapsto} A, Ta)$	// (17), (18)
20. $B \equiv Xa$	// (5), (19), belief rule

Results of Table II prove that  $B$  believes  $Xa$  to be true (belief (20)), that is, as we considered  $Xa$  to be comprised of @A,  $Na$ , and  $Ea$ ,  $B$  now believes all these statements to be true. With the belief (15),  $B$  believes  $\overset{Ka}{\mapsto} A$  to be true. Finally,

BAN logic proves that  $B$  believes simultaneously  $@A$ ,  $Na$ ,  $Ea$  and  $\stackrel{K_a}{\vdash} A$  to be true, and TAKES protocol is as such formally proved.

### B. Key Update/Revocation Messages

For key update or revocation, we redefine the statement  $Xa$  to be comprised of the new address of principal  $A$ , its former address, its new public key (if it is an update) or its old public key (if it is a revocation) and a start of the validity date (i.e. for the revocation/update message).  $Xa$  is part of the transmitted information and its definition serves only to condense the BAN formula.

In BAN logic, the key update and revocation message can be represented as follows:

$$\text{Link channel: } A \longrightarrow B : \{Xa, Ta\}_{K_{a-1}}$$

Again, several BAN logic assumptions (see Table III) must be provided.

TABLE III  
INITIAL ASSUMPTIONS FOR KEY REVOCATION OR UPDATE.

1.	$B \equiv \stackrel{K_a}{\vdash} A$
2.	$B \equiv \#(Ta)$
3.	$B \equiv A \Rightarrow Xa$

The formal verification of the message is given in Table IV. The conclusion of the verification is that  $B$  now trusts  $Xa$ . As such, the key update or revocation operation is formally proven to achieve the goals.

### VII. SECURITY ANALYSIS

This section discusses the protection mechanisms integrated into our solution. The attacker is behaving according to the Dolev-Yao model [23], that is, the attacker can eavesdrop, modify, replay or create any messages. The only one limitation is that the attacker can not break cryptographic protections (e.g. cannot fake a digital signature).

The message sent over the link channel during the public key distribution (message (2) of Figure 1) does not disclose any useful information to the attacker. The only sensitive information is the passphrase ( $secretA$ ) keying the HMAC but it can not be extracted from the message. Any attempt to tamper the message is detected during the digital signature

TABLE IV  
VERIFICATION STEPS FOR KEY REVOCATION AND UPDATE.

$A \longrightarrow B : \{Xa, Ta\}_{K_{a-1}}$
4. $B \triangleleft \{Xa, Ta\}_{K_{a-1}}$
5. $B \equiv A \vdash (Xa, Ta)$ // (1), (4), msg.-meaning rules
6. $B \equiv \#(Xa, Ta)$ // (2), freshness rule
7. $B \equiv A \equiv (Xa, Ta)$ // (5), (6), nonce-verif. rule
8. $B \equiv Xa$ // (3), (7), belief operator

verification. Also, replacing the public key is detected as it is breaking the HMAC verification.

The attacker might disturb transmissions over the ad hoc network so that link messages are dropped, leading to a denial of service attack. However, thanks to the OOB message, participants are warned on the intent of the sender to distribute its public key and the lack of incoming messages at the receivers will indicate a possible on-going attack. It is also possible that the attacker replays the link messages. These messages can be stored by the receivers, but they will not be processed until the corresponding OOB message is received. Upon receiving the messages, all duplicate messages are discarded, and hence, no extra resource consumption occurs.

If the order of the messages is not respected (i.e. the OOB message is received before the link message), an attacker can then learn the  $secretA$  before the link message is sent and he is then able to build valid link messages containing his own public key, a valid digital signature (computed over its private key) and a valid HMAC (containing its public key). Therefore, we stress that the correct ordering of the messages is essential for TAKES security.

TAKES, as it is currently defined, also relies on a crucial component, namely the CGA addresses. The CGA addresses are initially derived from the SUCV crypto-based identifier, therefore most of the literature on the SUCV applies to CGAs as well. In document [18], Montenegro and Castelluccia discuss the weaknesses of SUCV. Their conclusion indicates that theoretical attacks on SUCV will remain prohibitively complex over the next decades and hence do not affect TAKES.

### VIII. IMPLEMENTATION

The TAKES protocol and the corresponding graphical user interface are implemented using the Python language and the GTKBuilder toolkit. The implementation is currently limited to the Linux operating system, as it relies on NDprotector<sup>1</sup> for CGA generation and verification. TAKES is licensed under the GNU General Public License version 3 and is publicly available<sup>2</sup>.

We tested the implementation within a Mobile Ad Hoc Network (MANET) environment composed of nodes interconnected through heterogeneous link technologies (Ethernet cable and wireless link), relying on the stigmergic-based routing protocol B.A.T.M.A.N.<sup>3</sup> to ensure packet delivery. This choice is motivated by the support of multicast in B.A.T.M.A.N. and the quality of its native Linux in-kernel implementation.

As described in Section IV, after an initial setting of the configuration parameters (public key, user's name, equipment's name), the user is able to securely transmit his/her public keys to the other participants within the network. To launch the public key transmission over the link layer, the user is requested to enter his/her one-time passphrase through the

<sup>1</sup><http://amnesia.org/ndprotector/>

<sup>2</sup><http://gitorious.org/takes/>

<sup>3</sup>Better Approach To Mobile Ad hoc Networking - <http://www.open-mesh.org/>

dialog popup presented in Figure 3. In practice, this (link) message is multicasted through UDP messages.

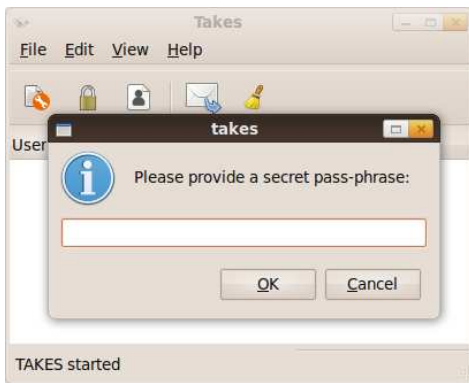


Fig. 3. Popup for launching TAKES public key broadcast.

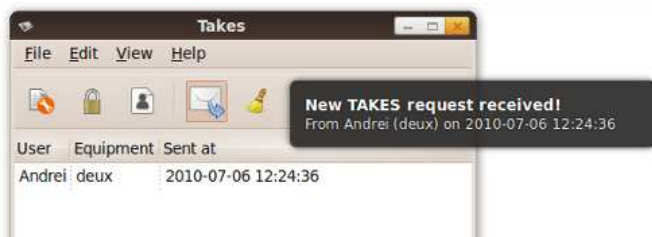


Fig. 4. Popup for incoming TAKES Messages.

As soon as other participants received a TAKES message, a notification is displayed on the screen (see Figure 4). Moreover, if several incoming TAKES messages are received they are appended to the list containing the not-yet-accepted messages sorted by chronological order. A double-click on one of these messages presents the user with the details contained in the message (source IPv6 address, hash of the message, public key in PEM format), and the option to either save the received public key in his/her local database or to remove the message from the list, e.g. in case he/she suspects that something is wrong. To acknowledge the message and save the public key, the user has to enter the owner's passphrase which is given to him/her through the OOB (after the link message was received).

The size of TAKES key broadcast message is 498 bytes for a 1024 bit long RSA key, and 761 bytes for a 2048 bit long RSA key. As cryptographic operation times are depending on available CPU resources on TAKES devices, it does not make sense measuring the operation times.

## IX. CONCLUSIONS AND PERSPECTIVES

In this paper, we presented a Trustful Authentication and Key Exchange Scheme (TAKES) adapted to ad hoc networks where no Trusted Third Party is available. Our proposal is a secure, reliable, and trustful key distribution mechanism which also serves to link identities to public keys. One of the very interesting features of TAKES is the simple-to-use Out-of-Band Channel (OOBC). The OOBC channel serves to divulge

a secret passphrase to all the participants so the authenticity of the link channel message (over the ad hoc network) is established. A high-level security is achieved as the trust in the message is conferred by personally trusting the participant divulging the passphrase.

Numerous tests have been successfully performed over a mobile ad hoc network based on B.A.T.M.A.N. tool. The results proved that TAKES performs as expected.

Future perspectives include improving implementation aspects, and also developing a modular system and a public API for security-enabled applications (e.g. securing routing protocols, VPNs, IPsec, etc.) in order to have an easy access to the locally stored information (for example the public key belonging to a specific user). Additionally, we will introduce application scenarios where TAKES is combined to existing security protocols and contributes to enhance the overall security level of the participants.

## ACKNOWLEDGMENT

This research is part of the TLCOM MobiSEND project financed by the French National Research Agency (ANR). Tony Cheneau accomplished this work during his PhD and was supported by a grant of TELECOM SudParis.

## REFERENCES

- [1] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, "A secure routing protocol for ad hoc networks," in *10th IEEE International Conference on Network Protocols*. IEEE Computer Society, 2002, pp. 78–89.
- [2] Y. Hu, D. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 1, pp. 175–192, 2003.
- [3] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," *Wireless Networks*, vol. 11, no. 1, pp. 21–38, 2005.
- [4] A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. Culler, "SPINS: Security protocols for sensor networks," *Wireless networks*, vol. 8, no. 5, pp. 521–534, 2002.
- [5] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 2002.
- [6] L. Zhou and Z. Haas, "Securing ad hoc networks," *Network, IEEE*, vol. 13, no. 6, pp. 24–30, 2002.
- [7] H. Deng, A. Mukherjee, and D. Agrawal, "Threshold and identity-based key management and authentication for wireless ad hoc networks," in *International Conference on Information Technology: Coding and Computing (ITCC04)*, vol. 1, 2004, pp. 107–115.
- [8] T. Aura, "Cryptographically Generated Addresses (CGA)," in *Information Security*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2003, vol. 2851, pp. 29–43.
- [9] A. Perrig and D. Song, "Hash visualization: A new technique to improve real-world security," in *International Workshop on Cryptographic Techniques and E-Commerce*, 1999, pp. 131–138.
- [10] E. Kim, W. Kong, and J. Yi, "Providing secure mobile device pairing based on visual confirmation," in *13th International Symposium on Consumer Electronics (ISCE'09)*. IEEE, 2009, pp. 676–680.
- [11] L. Feeney, B. Ahlgren, A. Westerlund, and A. Dunkels, "Spontnet: Experiences in configuring and securing small ad hoc networks," in *The Fifth International Workshop on Network Appliances (IWNA5)*. IEEE Computer Society, 2002.
- [12] D. Balanz, D. Smetters, P. Stewart, and H. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," in *9th Annual Network and Distributed System Security Symposium (NDSS)*, 2002, pp. 7–19.
- [13] N. Saxena and M. Uddin, "Device Pairing Using Unidirectional Physical Channels," in *Mobile and Wireless Networks Security: Proceedings of the MWNS 2008 Workshop*, 2008, p. 27.

- [14] R. Mayrhofer and M. Welch, "A human-verifiable authentication protocol using visible laser light," *The Second International Conference on Availability, Reliability and Security (ARES'07)*, pp. 1143–1148, 2007.
- [15] J. McCune, A. Perrig, and M. Reiter, "Seeing-is-believing: Using camera phones for human-verifiable authentication," in *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2005, pp. 110–124.
- [16] M. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, "Loud and clear: Human-verifiable authentication based on audio," in *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. IEEE Computer Society, 2006, p. 10.
- [17] W. Claycomb and D. Shin, "Secure device pairing using audio," in *43rd International Conference on Security Technology*. IEEE, 2009, pp. 77–84.
- [18] G. Montenegro and C. Castelluccia, "Crypto-based identifiers (cbids): Concepts and applications," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 97–127, 2004.
- [19] J. Arkko, J. Kempf, B. Zill, and P. Nikander, "SEcure Neighbor Discovery (SEND)," Internet Engineering Task Force, RFC 3971, Mar. 2005.
- [20] J. Laganier and G. Montenegro, "Using IKE with IPv6 Cryptographically Generated Address," Internet Engineering Task Force, Internet-Draft draft-laganier-ike-ipv6-cga-02, Jan. 2008.
- [21] J. Sheu, C. Chao, W. Hu, and C. Sun, "A clock synchronization algorithm for multihop wireless ad hoc networks," *Wireless Personal Communications*, vol. 43, no. 2, pp. 185–200, 2007.
- [22] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 1, pp. 18–36, 1990.
- [23] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 2002.