2014 IEEE 13th International Symposium on Network Computing and Applications

Formal Analysis of Secure Device Pairing Protocols

Trung Nguyen
TELECOM ParisTech
Computer Science and Networking Department
CNRS, LTCI, UMR 5141,
46 rue Barrault, 75013 Paris, France
Email: ttnguyen@telecom-paristech.fr

Jean Leneutre
TELECOM ParisTech
Computer Science and Networking Department
CNRS, LTCI, UMR 5141,
46 rue Barrault, 75013 Paris, France
Email: jean.leneutre@telecom-paristech.fr

***Abstract*—The need to secure communications between personal devices is increasing nowadays, especially in the context of Internet of Things. Authentication between devices which have no prior common knowledge is a challenging problem. One solution consists in using a pre-authenticated auxiliary channel, human assisted or location limited, usually called out-of-band channel. A large number of device pairing protocols using an out-of-band channel were proposed, but they usually suffer from a lack of formal analysis. In this paper, we introduce a formal model, conceived as an extension of Strand Spaces, to analyze such protocols. We use it to analyze a device pairing protocol with unilateral out-of-band channel proposed by Wong & Stajano. This leads us to discover some vulnerabilities in this protocol. We propose a modified version of the protocol together with a correctness proof in our model.**

I. Introduction

Securing the wireless communication channel requires to be able to establish an initial trust relation between unassociated devices. Such a trust initialization process is commonly called *Secure Device Pairing*, or *Secure Bootstrapping*, or *Secure First Connect*. Due to the heterogeneity of devices and lack of official standard, no existing security infrastructure or scheme could provide an universal solution. Unfamiliar devices with no common trust cannot take advantage from traditional cryptographic protocols (i.e. authenticated key exchange protocols) for this task when there does not exist any pre-shared secrets, or the authenticated public key of each other.

Trying to solve these problems, a great body of work proposes to introduce some form of human involvement in the secure pairing process. This human involvement is achieved by using an auxiliary channel between the devices that is both observable and controllable by the human that manages the devices. In this paper, we will use the general term out-of-band channel (OOB). A great number of technologies have been proposed to implement the OOB channel together with several device pairing protocols as documented in [1]. Security of these protocols is in general evaluated informally.

There is therefore a need for more formal validation. The usual verification approaches designed for classical authentication protocols usually employ an attacker model based on Dolev-Yao model [2]. In the case of secure device pairing protocol, since the OOB channel is under a restricted control by the user, the power of the attacker may be reduced with comparison to Dolev-Yao model during some steps of the protocols. Therefore traditional verification techniques are not directly applicable.

Our objective in this article is to define a formalism which models device pairing protocols in a natural manner, and permits verification of security properties relevant to these protocols. We conceive such a formalism as an adaptation of Strand Spaces [3]. The model of Strand Spaces is a flexible formalism which represents protocols as a set of local views of participants in a run of a protocol. Taking advantage of this flexibility, our modifications mainly consist in extending Strand Spaces to deal with OOB channels. In particular, the attacker model must be refined to take into account the different types of channels, i.e. unsecured channels and OOB channels.

In the rest of the paper, section II will sum up the main concepts of OOB based device pairing protocols and present some existing works on verification of secure device pairing protocols. Section III introduces our extension of Strand Spaces for secure device pairing protocols. Section IV describes the formal analysis of one secure device pairing protocol taken from [4]. Interestingly, our analysis points out some vulnerabilities on this protocol that were not noticed before to our knowledge. The last section concludes.

II. Related Works

A. Secure Device Pairing schemes using OOB Channels

Traditionally, the specification of an OOB based device pairing scheme contains two main elements: (i) the description of the pairing protocol and (ii) the assumptions on the OOB channel. The specification of the pairing protocol will detail in particular which steps require an OOB channel.

An OOB channel provides an initial security level. The adversary abilities on these channels are limited to a subset of previous malicious actions possibly empty (depending on the type of the OOB channel). An OOB channel provides at least the first two of the following security objectives: entity authentication, data integrity, and data confidentiality. Three main types of OOB channels are usually distinguished: the *private channel* ensuring all security objectives, and both the *protected channel* and the *public channel* just ensuring authentication and integrity. The difference between three types of OOB channels and penetrator power for each kind is summarised in table I

Reference [5] studied techniques which enable wireless devices to authenticate together over an unsecured channel with a human assistance. The main idea is that a user copies a data output from one device to the other (Output/Input),

978-1-4799-5393-6/14 $31.00 © 2014 IEEE
DOI 10.1109/NCA.2014.50

or compare the output of two devices (Output/Output) or enters same data into both devices (Input/Input). However, these techniques may suffer from a lack of ease of use when the length of data is too long. A simple solution consists in truncating the hash value to 16 or 32 bits, but in this case an adversary may still be able to launch a MitM attack by finding collisions on the first bits of the hash values.

The authors of [5] introduced three types of manual authentication protocols to offer a proper trade-off between security and usability: MANA I for Output/Input, MANA II for Output/Output, and MANA III for Input/Input. These protocols practically reduce the bandwidth of OOB channel to k-bits (16-20 bits) for each direction while the MitM attack success probability is limited to 2^{-k}. MANA-based protocol family usually requires a strong assumption on the OOB channel: an adversary is not able to delay or relay any OOB message. Getting rid of this strong assumption, [6] proposed a protocol based on Short Authentication String (SAS) of length 15 bits, while still preserving the 2^{-k} attack success probability.

Readers eager to learn more about device pairing protocols can consult the recent review on the topic [1].

B. Formal verification of secure device pairing protocols

The work presented in [7] tried to answer to the following question: are auxiliary channels necessary to provide authentication without pre-shared knowledge? Using BAN logic [8], they prove that device authentication using a single channel is not possible . From this analysis, they propose an extension of BAN logic taking into account OOB channels, and using this extension the *Talking to Strangers* protocol from [9] and a simplified version of Wong-Stajano protocol [4] were shown to be correct. However, as we will see in the next section IV, the Wong-Stajano protocol is vulnerable to an attack. In fact, the proposed formalism does not offer enough expressivity to correctly model the Wong-Stajano protocol.

Formal verification of specific versions of Bluetooth protocols has received a lot of attention in the literature. Several proposals were introduced to take into account Bluetooth security weaknesses from the old version 2.0 to the brand new version 4.0. Some verification tools have been applied such as ProVerif in[10], and PRISM probabilistic model checker in [11]. These works are a first steps towards an automated analysis of formal model of human-assisted protocols.

III. Extended Strand Spaces with out-of-band channels

Due to lack of place, we do not recall here the whole theory of Strand Spaces, but focus on the extensions necessary to examine secure pairing protocols based on Diffie-Hellman scheme. For a complete background on Strand Spaces the reader can consult [3], [12], and [13]. The extensions mainly concern the algebra and the penetrator model.

TABLE I: Comparison of Out-of-Band Channels

Out of Band Channel		Adversary Power				
Type	Interface	Overhear	Block	Delay	Relay	Forge
Private	Cable	Ø	Ø	Ø	Ø	Ø
Protected	Button, Accelerometer	√	√	Ø	Ø	Ø
Public	LED, Speaker	√	√	√	√	Ø
Insecure	Wireless	√	√	√	√	√

Before presenting our extension of Strand Spaces, we formulate some supplementary assumptions concerning the execution of device pairing procedures, that we will have to take into account.

A. Model assumptions

We now make several practical assumptions in our model as follows:

- The hash functions used in the secure device pairing protocol are perfect, that is the attacker cannot perform with success the following attacks: collision attack, pre-image attack, and second-image attack.
- There is no more than one instance of a particular role sending/receiving on an OOB channel at a given time.
- When one device sends the Accept/Reject information, the other device confirms this decision.
- After a device pairing procedure, the communication session will start later. But in case of no evidence of exchanging procedure, the device pairing procedure replays again with a new session.

B. Extension to the Algebra

Our definition of Strand Space algebra is based on the definition from [13], which adds to model the possibility to deal with DH operation, hash functions, and signatures. To take into account device pairing protocols, we do not need to consider signatures (neither asymmetric encryption), but must add keyed hash function, or MAC function. We thus redefine the set of terms as follows:

Definition 3.1: The set of *terms* $\mathcal{A}$ is assumed to be freely generated from four disjoint sets: predictable texts $\mathcal{T}$, unpredictable texts $\mathcal{R}$, keys $\mathcal{K}$, and Diffie-Hellman values $\mathcal{D}$.

The set of keys $\mathcal{K}$ is divided into two disjoint sets: verification keys $\mathcal{K}_{Ver}$, and keys for symmetric encryption $\mathcal{K}_{Sym}$.

Compound terms are built by these operations:

- join: $\mathcal{A} \times \mathcal{A} \rightarrow \mathcal{A}$, which represents concatenation of terms.
- encr: $\mathcal{K}_{Sym} \times \mathcal{A} \rightarrow \mathcal{A}$, which represents encryption.
- DH: $\mathcal{D} \times \mathcal{D} \rightarrow \mathcal{D}$, which represents the Diffie-Hellman operation. We denote the range of DH by $\mathcal{D}_{DH}$.
- hash: $\mathcal{A} \rightarrow \mathcal{K}_{Sym}$, representing hashing into keys. We denote the range of hash by $\mathcal{K}_{hash}$.
- MAC: $\mathcal{K}_{Sym} \times \mathcal{A} \rightarrow \mathcal{K}_{Sym}$, representing MAC operation with a key into keys.

Terms will be denoted by t, t' possibly indexed by an integer. The elements from the set of unpredictable or random texts $\mathcal{R}$ are used to play the role of nonces in protocols and will be denoted by r possibly indexed with the identifier of

an agent. The elements of $\mathcal{K}$ (resp. $\mathcal{D}$) will be denoted by k (resp. d) possibly indexed by an identifier (resp. integer). In the following, $encr(k,t)$, $hash(t)$ and $MAC(k,t)$ will be respectively noted $\{|t|\}_k$, $h(t)$ and $h_k(t)$. The term $join(t,t')$ will be noted t,t' or (t,t') when necessary to avoid confusion.

In our extension, we will need to explicitly distinguish between different channels. We thus need to define what is a channel.

Definition 3.2 (Channel): A *channel* is a group of devices which can exchange messages in the same region.

One device may use more than one channel. For example, given two channels ch_1 and ch_2 and 3 devices A, B, C, the devices A and B may use ch_1, whereas B and C use ch_2.

If no supplementary assumptions are declared, a channel is by default an unsecured public wireless channel. Any specific assumption on an OOB channel, must be specified before formalizing the protocol. An OOB will be usually limited to two devices.

Since a protocol may use several channels, when sending or receiving a term, the used channel must be specified. The definition of signed term is modified in consequence.

Definition 3.3 (Signed term): A *signed term* is a triplet $\langle \delta, t, ch \rangle$, noted $\delta_{ch}t$, where δ is $+$ (sending) or $-$ (reception), t a term, and ch the channel on which t is sent or received.

Actually, we will see in subsection III-C that the terms manipulated by a penetrator may receive another sign. By convention, we will specify the channel only when using an OOB Channel: $-_{ch}t$ means that the term t is received on the OOB channel ch, and $-t$ will denote the reception of t on the public wireless channel.

Based on this new definition of signed terms, the definitions of *strand space*, *node*, *edge*, *originating term*, *uniquely originating term*, and *bundle*, *height of a strand* are the same than in [3].

We refine the notions of *subterm* and *component* from previous works on Strand Spaces as follows.

Definition 3.4 (Subterm): We say that t is a *subterm* of t', written $t \sqsubset t'$ if:

- $t = t'$ or
- $t' = (t'_1, t'_2)$ then $t \sqsubset t'_1$ or $t \sqsubset t'_2$,
- if $t' = \{|t''|\}_k$, then $t \sqsubset t''$,
- if $t' = h(t'')$, then $t \sqsubset t''$,
- if $t' = h_k(t'')$, then $t \sqsubset t''$,
- if $t' = DH(d_1, d_2)$, then $t \sqsubset d_1$ or $t \sqsubset d_2$

Definition 3.5 (Component): We say that a term t is a *component* of term t', written $t \sqsubset_c t'$, if t' can be obtained by concatenating t with others terms.

For example, the term $(A, g^a, h(A, g^a)$, where g^a denotes a Diffie-Hellman value, contains three components: A, g^a, and $h(A, g^a)$.

At last, we introduce the notion of boxed term.

Definition 3.6 (Boxed term): For a given bundle, we say that a term t is *boxed* at node n, if there exists terms t' and t'' such that $t \sqsubset t'$, $t' \sqsubset term(n)$, and t' has one of the following forms: $\{|t''|\}_k$, $h(t'')$, $h_k(t'')$.

C. Extended Penetrator Model

The new penetrator model must take into account the different kind of channels used in the secure device pairing protocols.

Concerning wireless channels, the original Dolev-Yao model is broadened with DH, hash and MAC operations as following:

- **F**. Fresh DH value: $\langle +d \rangle$ where $d \in \mathcal{D}_P$ with $\mathcal{D}_P \subset \mathcal{D}$ and $\mathcal{D}_P \cap \mathcal{D}_{DH} = \emptyset$
- **H**. Hashing: $\langle -t, +h(t) \rangle$
- **MAC**. MAC: $\langle -t, -k, h_k(t) \rangle$

As described in subsection II-A, the penetrator power is intentionally limited in term of message manipulation on OOB channels. He is prevented from performing actions on private OOB channels, however he is still able to realize some actions on public or protected OOB channels. We therefore need specific strands to model the actions of the penetrator these latter types of OOB channels. To do so, we extend the signed terms with a new event, $\#_o t$, meaning that the penetrator suspends the message t on OOB channel o. This brand new event is only adopted for public or protected OOB. Consequently, we extend the penetrator model with the following two penetrator traces on OOB channels:

- **SUS**. Suspending Message: $\langle -_o t, \#_o t \rangle$ where o is an OOB channel of type public or protected.
- **REL**. Releasing Message: $\langle \#_o t, +_o t \rangle$ where o is an OOB channel of type protected.

The dropping attack can be modeled by SUS strand without the REL strand. Moreover, REL strand only works for a term t over a public OOB channel o when there exists a SUS strand for t over o.

Having defined the penetrator model, we can now define the notion of revealed term.

Definition 3.7 (Revealed term): For a given bundle, a term t is called to be $revealed$ at node n if:

- $t \sqsubset term(n)$, and t can be obtained by the penetrator using his knowledge at node n, and
- for any n' that precedes n ($n' \preceq n$) such that $t \sqsubset term(n')$ the penetrator cannot obtain the t using his knowledge at node n'.

D. Security Properties

Intuitively, the goal of secure pairing device protocols is to guarantee that two devices with no prior shared knowledge and sharing a common OOB channel, receive the same agreement dataset after the acceptance notification. To formalize the corresponding security property, we adapt the definition of *agreement property* from [14] to our situation.

Definition 3.8 (Agreement Property): We say that a protocol guarantees an initiator A *agreement* with a responder B on a set of data items ds, if whenever A (acting as initiator) completes a run of the protocol, apparently with responder B, then B has previously been running the protocol acting as a responder apparently with A, and the two agents received the same ds at the end of a run.

The penetrator can attack the protocol if at the end of its run, both devices reach to Accept state, yet having a different agreement dataset.

IV. Analysis of Wong-Stajano Protocol

This section applies the previously presented model to to analyses Wong-Stajano Protocol with Unidirectional Channel. Wong and Stajano proposed in [4] a new mutual authentication and key agreement protocol over bidirectional and unidirectional authenticated channels. The authenticated channels ensure data origin authenticity but does not provide confidentiality. Their protocols exploited a short authenticated string over visual channels which provides integrity and data origin authenticity. The Wong-Stajano (WS) Protocol with unidirectional channel is presented in figure 1. Its model in our extension of Strand Spaces is defined below.

Definition 4.1: An infiltrated strand space $(\Sigma, \mathcal{P})$ is a Wong-Stajano protocol space if Σ is the union of three kinds of strands:

- Penetrator strands $s \in \mathcal{P}$,
- "Initiator strand" with trace $Init[r_B, k_B, g^a, g^b]$ defined to be:
  $\langle +g^a, -(B, g^b, h_{k_B}(B, g^b, g^a, r_B), +_o ACK, -_o r_B, -k_B \rangle$,
  where $B \in \mathcal{T}_{name}$, $ACK \in \mathcal{T}$, and $g^a, g^b \in \mathcal{D} \backslash \mathcal{D}_P$,
- "Responder strand" with traces $Resp[r_B, k_B, g^a, g^b]$ defined to be:
  $\langle -g^a, +(B, g^b, h_{k_B}(B, g^b, g^a, r_B)), -_o ACK, +_o r_B, +k_B \rangle$,
  where $B \in \mathcal{T}_{name}$, $ACK \in \mathcal{T}$, and $g^a, g^b \in \mathcal{D} \backslash \mathcal{D}_P$,

with o a public OOB channel.

Unfortunately, the agreement property does not hold for the Wong-Stajano protocol.

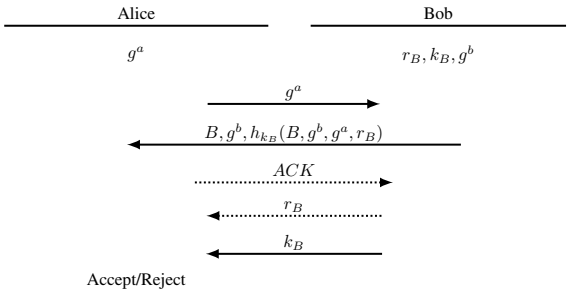

Fig. 1: Wong-Stajano protocol with unidirectional channel

1) Responder's Guarantee for Wong-Stajano protocol: Responder's guarantee for Wong-Stajano protocol is stated as follows:
Let $\mathcal{B}$ be a bundle containing a strand st' in $Resp[r_B, k_B, g^a, g^b]$ of height 5. If st' uses a public OOB channel o in $\langle st', 3 \rangle$ and $\langle st', 4 \rangle$, and r_B, g^b, k_B are uniquely originating on st', then $\mathcal{B}$ contains a unique strand st in $Init[r_B, k_B, g^a, g^b]$ of height 5 that also uses the channel o. Moreover, both strands agree on g^a, g^b.

Proving the responder's guarantee requires to prove the following lemmas.

Lemma 4.2: r_B uniquely originates on $\langle st', 2 \rangle$

Proof: Providing that r_B is uniquely originating on Σ, node $\langle st', 2 \rangle$ is a positive node, and $r_B \notin K$, thus the only possibility is that r_B uniquely originates at $\langle st', 2 \rangle$. ■

Lemma 4.3: There exists a regular node n_4 in an initiator strand such that $term(n_4) = -_o r_B$.

Proof: Since the responder strand receives the acknowledgement from Initiator, the initiator needs to have a regular node with term $-_o r_B$. Attacker cannot forge this acknowledgement because it is transmitted ob the OOB channel. ■

Lemma 4.4: There exists a regular node n_3 in an initiator strand such that $term(n_3) = -_o ACK$.

Proof: Because the term of node $\langle st', 3 \rangle$ is $-_o ACK$, there is some initiator strand which has a node n_3 which term is term $+_o ACK$. The node n_3 exploits the same OOB channel o than $\langle st', 3 \rangle$. ■

Then, to complete the proof of responder's guarantee property, we should prove that:

There exists a regular node n_2 in an initiator strand such that $term(n_2) = -(B, g^b, h_{k_B}(B, g^b, g^a, r_B)$.

To prove this, we should show that the term of node n_2 cannot be sent from a penetrator strand. It easy to check for one of the following strands: ***M, R, S, K, E, D, F, H, MAC***. However we cannot conclude with the ***C*** strand.

Indeed, using the ***C*** strand, the attacker may send $(B, g^x, h_{k_B}(B, g^x, g^a, r_B))$ to the initiator strand. It supposes that he used before the ***MAC*** strand and that he knows k_B, and r_B which are normally sent after node $\langle st', 4 \rangle$. The attacker may have learnt these values in a previous session. Let suppose that in a previous session, the attacker applies the strand $SUS = \langle -_o r_B, \#_o r_B \rangle$ to delay the delivery of message to the initiator strand, and then receive k_B. If the initiator does not receive the value r_B before expiration time, he will restart a session according to assumption. In current session, after sending $(B, g^x, h_{k_B}(B, g^x, g^a, r_B))$, the attacker can use $SUS = \langle -_o ACK, \#_o ACK \rangle$ to drop the ACK message on the OOB channel sent by the initiator. The attacker then execute $REL = \langle \#_o r_B, +_o r_B \rangle$ to deliver the message r_B to the initiator strand. Consequently, the responder's guarantee for Wong-Stajano protocol is not satisfied. Finally, after receiving the r_B message, the new initiator strand verify MAC value in node n_2, then sends the Accept. The attack is successful. Finally, the responder cannot guarantee for a regular initiator strand.

The complete attack scenario is described in table II.

TABLE II: ATTACK SCENARIO AGAINST RESPONDER'S GUARANTEE IN WONG-STAJANO PROTOCOL

Step 1.1	Initiator sends g^a on wireless channel
Step 1.2	Responder replies with $(B, g^b, h_{k_B}(B, g^b, g^a, r_B))$ on wireless channel
Step 1.3	Initiator sends ACK on OOB channel
Step 1.4	Attacker suspends r_b sent by Responder over the OOB channel
Step 1.5	Attacker intercepts k_B sent by Responder on wireless channel
Step 1.6	Initiator waits for r_B until expiration time and starts a new session
Step 2.1	Attacker intercepts $g^{a'}$ sent by Initiator on wireless channel
Step 2.2	Attacker sends $(B, g^x, h_{k_B}(B, g^x, g^{a'}, r_B))$ on wireless channel
Step 2.3	Attacker drops ACK sent by Initiator on OOB channel
Step 2.4	Attacker releases r_B, suspended at step 1.4, on the OOB channel
Step 2.5	Attacker sends k_B, intercepted at step 1.5, on OOB channel
Step 2.6	Initiator accepts the execution of the second session, believing he is sharing a fresh session key with responder, known actually by the attacker

TABLE III: ATTACK SCENARIO AGAINST INITIATOR'S GUARANTEE IN WONG-STAJANO PROTOCOL

Step 1.1	Attacker intercepts g^a sent by Initiator on wireless channel
Step 1.2	Attacker replies with $(B, g^x, h_{k_X}(B, g^x, g^a, r_X))$ on wireless channel
Step 1.3	Attacker suspends ACK sent by Initiator on OOB channel, and starts a new session with Responder
Step 2.1	Attacker sends $g^{x'}$ on wireless channel
Step 2.2	Responder sends $(B, g^b, h_{k_B}(B, g^b, g^{x'}, r_B))$ on wireless channel
Step 2.3	Attacker releases ACK sent by Initiator at step 1.3
Step 2.4	Attacker suspends r_B sent by Responder on OOB channel
Step 2.5	Responder intercepts k_B sent by Responder on Wireless channel
Step 2.6	At the end of the execution, Responder believes he shares a fresh session key with Initiator, known actually by the Attacker

2) Initiator's Guarantee for Wong-Stajano protocol: Initiator's guarantee for Wong-Stajano protocol is stated as follows:
Ley $\mathcal{B}$ be a bundle containing a strand st in $Init[r_B, k_B, g^a, g^b]$ of height 5. If st uses a public OOB channel o in $\langle st, 3\rangle$ and $\langle st, 4\rangle$, and g^a is uniquely originating on st, then $\mathcal{B}$ contains a unique strand st' in $Resp[r_B, k_B, g^a, g^b]$ of height 5 that also uses the channel o. Moreover, both strands agree on g^a and g^b.

As for the responder's guarantee, the initiator's guarantee does not hold for Wong-Stajano protocol. Trying to prove it leads to the attack scenario detailed in table III.

V. CONCLUSION

In this paper, we extended the original strand space model to be able to analyze secure device pairing protocols. To achieve this, we modified the model so that it becomes possible to take into account protocols using several kind of channels, including OOB channels. The penetrator model has been adapted in consequence. This extension was used to formalize and analyze the Wong-Stajano mutual authentication protocol with unilateral OOB channel. It successfully pointed us some flaws in the Wong-Stajano protocol that have never been noticed before to our knowledge.

Aforementioned works on this topic, mainly apply existing verification tools initially. We rather chosen to define a dedicated formalism able to model the specificities of device pairing protocols in a natural manner, and the results obtained so far seems promising. Concerning future work, we will first try to extend the model in order to capture a broader class of secure device pairing protocols such as Short Authentication String (SAS) based protocols. In a second step, we plan to study how we could automate the analysis procedure.

REFERENCES

[1] S. Mirzadeh, H. Cruickshank, and R. Tafazolli. Secure device pairing: A survey. *Communications Surveys Tutorials, IEEE*, 16(1):17–40, First 2014.

[2] D. Dolev and Andrew C. Yao. On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208, Mar 1983.

[3] F.J. Thayer Fabrega, J.C. Herzog, and J.D. Guttman. Strand spaces: why is a security protocol correct? In *Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on*, pages 160–171, May 1998.

[4] Ford Long Wong and Frank Stajano. Multichannel security protocols. *IEEE Pervasive Computing*, 6(4):31–39, 2007.

[5] C. Mitchell, C. Gehrmann, and K. Nyberg. Manual authentication for wireless devices. *Cryptobytes*, January 2004.

[6] Serge Vaudenay. Secure communications over insecure channels based on short authenticated strings. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 309–326. Springer Berlin Heidelberg, 2005.

[7] William Claycomb and Dongwan Shin. Extending formal analysis of mobile device authentication. *Journal of Internet Services and Information Security (JISIS)*, 1(1):86–102, 5 2011.

[8] Michael Burrows, Martn Abadi, and Roger Needham. A logic of authentication. *ACM TRANSACTIONS ON COMPUTER SYSTEMS*, 8:18–36, 1990.

[9] Dirk Balfanz Smetters, Dirk Balfanz, D. K. Smetters, Paul Stewart, and H. Chi Wong. Talking to strangers: Authentication in ad-hoc wireless networks. 2002.

[10] Richard Chang and Vitaly Shmatikov. Formal analysis of authentication in bluetooth device pairing. In *Proc. of LICS/ICALP Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA)*, Wroclaw, Poland, July 2007.

[11] Marie Duflot, Marta Kwiatkowska, Gethin Norman, and David Parker. A formal analysis of bluetooth device discovery. *International Journal on Software Tools for Technology Transfer*, 8(6):621–632, 2006.

[12] Joshua D. Guttman and F. Javier Thayer. Authentication tests and the structure of bundles. *Theor. Comput. Sci.*, 283(2):333–380, June 2002.

[13] J.C. Herzog. The diffie-hellman key-agreement scheme in the strand-space model. In *Computer Security Foundations Workshop, 2003. Proceedings. 16th IEEE*, pages 234–247, June 2003.

[14] Gavin Lowe. A hierarchy of authentication specifications. In *Computer Security Foundations Workshop, 1997. Proceedings., 10th*, pages 31–43, Jun 1997.