# ISO MANA Certificates in Practice

Kaisa Nyberg
Department of Information and Computer Science
Aalto University School of Science and Technology, and
NOKIA, Finland
kaisa.nyberg@tkk.fi

## ABSTRACT

The international standard ISO 9798-6 specifies MANA protocols for data authentication and an application of the MANA I protocol to public key authentication and key exchange. The resulting protocol is called the MANA certificate protocol an can be viewed as a passkey based key agreement protocol. In this paper we present an application of MANA certificate protocol to a wireless security association protocol and present a practical security analysis by quantifying the achieved security level in terms of the length and structure of the passkey.

## Keywords

wireless security association, manual authentication, passkey-based key agreement

## 1. INTRODUCTION

A *security association* specifies the cryptographic keys and algorithms to be used for secure communication among the participants in the association. Setting up a security association for a pair of entities over a wireless connection typically rely on the user to transmit security related information between the entities. Various association models have been specified for short range communication technologies such as Bluetooth, Wi-Fi and Wireless USB. One approach is to use auxiliary communication channels like near field communication, infrared, audio or video channels. When such auxiliary channels are not available, some form of user action, such as entering a passkey or verifying a checksum, is needed. In the interest of usability and security, user actions should be kept as simple as possible [11].

In addition to the traditional methods using shared secret passkey, new protocols based on checking a short string have been introduced. The first such protocol was developed by P. Zimmermann for PGPfone [15] and recently, further developments were presented by S. Vaudenay [13] and S. Laur, et al. [4], [5]. This approach, known as the numeric comparison method, was considered very promising [3] and was adopted for the Wireless USB as an alternative to a more conventional cable-based association model [9]. In spite of its benefits, the numeric comparison method has not gained unreserved confidence of manufacturers and implementers, mainly because it requires a different mental model than the traditional passkey-based methods. This is one reason why Bluetooth SIG is providing both passkey-based and numeric comparison association methods in the Secure Simple Pairing specification [2]. An extensive analysis of contemporary security association methods is presented in [1], see also [7].

Currently the Wireless USB association supplement specification provides a public key exchange method using numeric comparison only. In this paper we present a passkey based key agreement protocol for Wireless USB. The protocol to be considered is the MANA certificate protocol [6] which is a passkey based authentication protocol and sets the communicating parties in different roles similar to the client-server setting in the TLS. The main distinctive feature is that the passkey cannot be selected or generated from scratch, but it is computed based on a short secret key and the public key of the entity. We also present a first practical analysis of the MANA certificate protocol and give some guidelines how to select the parameter lengths to achieve, in the context of Wireless USB, a security level comparable to the one offered by the current numeric comparison association model.

The rest of the paper is organised as follows. We start by giving background of the Wireless USB communication architecture and security and discuss the practical shortcomings of the numeric comparison method In Section 2. In Section 3, we present the MANA certificate protocol as specified in [6]. In Section 4, we describe an application of ISO MANA certificate protocol to Wireless USB association models and outline a new fixed passkey association model. In Section 5 we analyse how the parameter lengths affect the security level on the host and the device side and discuss some further security properties of the fixed passkey association method. We conclude in Section 6.

## 2. WIRELESS USB ASSOCIATION MODELS

### 2.1 WUSB Connections

**Wireless USB (WUSB)** is a short-range wireless communication technology for high speed data transmission. The WUSB communication architecture is asymmetric and makes a distinction between the two communication parties: one is called *host* and another is *device*. The WUSB device is typically an entity that provides some service while the hosts are entitiess using the service. Examples of WUSB devices include wireless beamers, printers, hard discs etc. When requiring a service a WUSB host initiates a connection with a WUSB device providing the service.

**User conditioning** is a process where the user grants permission to the host and the device to allow a new association to be established. Conditioning is required to prevent accidental association when there are multiple active hosts and

devices within the range. The actual contioning steps taken by the user are different for each association method and are used to indicate which association model to use. User conditioning has implications to the security properties of the association models and its correct implementation is critical to the security.

## 2.2 Association Models 1.0

WUSB Association Models Supplement 1.0 specification [9] supports two association models:

**Cable model** uses key transfer from host to device over a wired USB connection. User conditioning reduces to the action of connecting the cable to the host and device.

**Numeric model** relies on the users to authenticate the Diffie-Hellman key agreement by comparing short integrity checksum values. The protocol was developed by Philip Zimmerman and was first used for the PGP phone [15] in establishing a shared secret key for encrypting the PGP phone call. For WUSB the check is based on 2-digit or 4-digit integer depending on the capability of the device's display. The protocol is similar to the one presented by S. Vaudenay in [13]. In Vaudenay's protocol the commitments are computed from fresh random values, whereas in WUSB numeric association model the commitments are computed from the public Diffie-Hellman values. This implies, in particular, that in WUSB numeric association, each run of the protocol requires the use of fresh Diffie-Hellman keypairs. A formal security proof of the WUSB numeric protocol was given in [5]. To start the association user conditions on the host and the device. The protocol runs and eventually the host and the device display each an integer to the user. The user compares the values on the displays and if the values are the same, conditions again on the host and the device. If the values are different the user rejects the association on both the host and the device.

A WUSB host is required to support all association models, while for a device a vendor is allowed to choose the model which suits the device best.

## 2.3 The Numeric Model in Practice

The numeric association model was the first key agreement protocol based on public key cryptography standardised for wireless local communication systems. Expectations were high as one can see from the presentation by Preston Hunt at WUSB conference in May 2005 although people were somewhat concerned about CPU power and memory required for handling 3072-bit modular arithmetic. Also the requirement of a display on all devices was a concern [3].

The 3072-bit MODP group #15 was selected to confirm with 128-bit symmetric key security, which is expected to offer adequate security far beyond the year 2030. In comparison, the upperbound of the active attacker's success probability, 1/100 or 1/10000 (depending on the length of the displayed integer) may look astonishingly large.

One interpretation of such parameter choices is that passive eavesdropping attacks are considered a far greater threat than active attacks. Already the plain Diffie-Hellman key exchange with sufficiently large keys protects against passive attacks. Due to user conditioning, active attacks can be launched only when an association setup is ongoing between the host and device. Under an active attack the numeric comparison will fail a number of times and each time the user must start a new association on both the host and device. Considering the time taken by each association attempt this level of protection against an active attacker has been adopted as sufficient.

The main drawback, possibly underestimated at the time when numeric comparison was introduced to WUSB, is the number of user actions it takes and the number of ways they can go wrong. Already early user studies on user-assisted authentication methods [12] suggested that numeric comparison is prone to the following *fatal error*: user confirms the association even if the displayed numbers are different. In general, the step of accepting or rejecting after numeric comparison is considered counterintuitive to the users and also difficult to get correct by the developers. For example, there exist implementations which do not give the user any explicit means to reject the association in case the displayed numbers are different. Instead, they rely on the user just to wait for time-out, see for example, [10], Question 36.

The numeric method requires more complex user interaction than the fixed passkey method. If passkeys were used, it would be sufficient to condition the host and the device only at the start the association by inserting the passkey. Some standards for wireless connectivity already provide passkey association methods based on Diffie-Hellman key exchange and shared secret passkeys, see for example, the Wi-Fi Protected Set Up [14] and the Passkey Entry method of the Bluetooth Simple Secure Pairing [2]. But these protocols do not support reusable, fixed passkeys. The security against active attacks is lost if the passkey is used multiple times.

Secure fixed passkey based key exchange schemes are abundant in the cryptographic literature. Unfortunately, many of them are patented by third parties or have unclear patent situation, which poses an unacceptable risk to industry, see for example [10], Question 21. Naturally, it would be a significant advantage if the new association method would be as similar to the Diffie-Hellman based numeric method as possible, so that existing functions and data frames specified for the numeric method could be reused as such or with minor changes. The ISO MANA certificates [6] offers one solution which might fullfil these requirements.

# 3. THE MANA CERTIFICATE PROTOCOL

A MANA certificate is a method for data authentication specified in Section 6.2.3 of [6]. Since this document is not publicly available we give a sufficiently detailed description here. A MANA certificate has two components, a key and a check value. The check value is computed from the key and the data. An example how the check value can be constructed using a universal hash family is given in the Annex C of [6], where also exact values of the upperbounds of the forgery probabilities are provided.

The idea is simple. Entity A has data $D$ that needs to be, or has already been sent to entity $B$. Then the following three steps are executed.

1. Entity $A$ generates a random key $K$, where $K$ is suitable for use with the check-value computation function $f$ shared by the two devices. Using this key $K$, entity $A$ computes a check-value $CV$ as a function of the data $D$, that is, $CV = f(D, K)$. The manual authentication certificate consists of $K$ and $CV$.

   The manual certificate $(K, CV)$ is then output to the user by the output interface of device $A$. The user reads the output certificate.

2. The user enters the manual certificate $(K, CV)$ to the input interface of entity $B$. The certificate $(K, CV)$ is stored in $B$.

3. When $A$ has sent data $D$ to $B$ over insecure channel, $B$ can verify the authenticity of the received copy $D'$ of the data $D$ using the stored values of $K$ and $CV$. Entity $B$ uses the key $K$ to recompute the check-value $CV' = f(D', K)$ as a function of the received data $D'$. If $CV = CV'$ then entity $B$ accepts the data $D' = D$ and outputs a success signal to the user. Otherwise it gives a failure signal.

In Annex A.3 of [6] an example is provided of how manual authentication certificates can be used to establish a shared secret key between two devices. The idea is the same as in the standard authenticated TLS handshake protocol [8]: The public key of the entity $A$ is authenticated using a PKI certificate after which $B$ authenticates the Diffie-Hellman key generated by $A$ and $B$ to $A$ based on a fixed short shared secret by $A$ and $B$ using a computationally secure check function. In manual authentication the PKI is replaced by a short MANA certificate.

When applying this protocol in practice for authentication of a Diffie-Hellman key exchange the following functions need to be specified:

- the check value computation function $f$ for manual certificate, and
- the computationally secure check function $g$.

For the latter, Annex A.3 of [6] proposes encryption of the password using the Diffie-Hellman key. In this paper, we use, instead of encryption, a computationally secure message authentication code.

MANA certificate has one drawback, the length of the passkey composed of the MANA certificate, is not optimal but twice the optimum length. That is, to achieve upperbound $2^{-\ell}$ to the forgery probability, the certificate must be at least $2\ell$ bits [1]. On the other hand, the passkey based protocol for Wi-Fi Secure Setup requires the same length to achive the same security level, but the passkeys can be used only once. Hence the MANA certificate protocol offers an equally secure but a more usable alternative to the existing standard security protocol for wireless associations.

# 4. APPLICATION OF MANA CERTIFICATES TO WUSB

In this section we present an application of the MANA certificate protocol given in Annex A.3 of [6] to WUSB association model. The protocol steps are depicted in Figure 1. The steps are divided to two stages, the offline stage where the fixed passkey is generated for the device and given to the user and the online stage where the association and connection context is formed.

The association protocol uses device specific fixed passkeys. Then in the notation of Section 3, a WUSB device acts as the entity $A$ and the WUSB host acts as entity $B$. The device specific MANA certificate constitutes the fixed passkey.

**PROTOCOL**

**Stage 1:** Manufacturer of the fixed passkey

1. Generates a fresh random private Diffie-Hellman exponent $F$ and computes $PK = g^F \bmod p$.
2. Generates a random integer $K$ of length $k$ digits.
3. Computes $f(PK||K)$ to produce a $c$-digit check value $CV$.
4. Stores $F$, $PK$ and $K$ in the device for further use by the association protocol.
5. Writes the $(k + c)$-digit integer $K||CV$ on a tag which is given to the user with the device.

**Stage 2:** Establishing a new association

1. The host and the device start the association.
2. User reads the string $K||CV$ from the tag of the device and enters it to the host.
3. Device sends $PK$ to the host.
4. Host computes $c$ digit check value $f(PK||K)$ and verifies that the result matches with $CV$. Host aborts the association if the verification fails.
5. Host generates a fresh random secret $B$ and computes $PK_H = g^B \bmod p$.
6. Host computes the shared secret $S = h(PK^B \bmod p)$.
7. Host computes $PC_H = g(S||K)$.
8. Host sends $PK_H$ and $PC_H$ to the device.
9. Device computes the shared secret $S = h(PK_H^F \bmod p)$.
10. Device computes $g(S||K)$ and verifies that it is equal to $PC_H$. Device aborts the association if the verification fails.

**Figure 1: Fixed Passkey Association Protocol**

In addition to the short check value computation function $f$ and the computationally secure check function $g$ introduced in Section 3 one additional function $h$ is needed. It is a key derivation function that is used for computing the shared secret association key $S$ from the "raw" Diffie-Hellman key. For example, if SHA-256 is available, both $g$ and $h$ can use SHA-256. Also the check value computation function $f$ can be based on SHA-256 by converting the SHA-256 output to an integer and truncating the result to a suitable length. The WUSB Association Model v 1.0 uses a similar function to compute the numeric check value in the numeric comparison association method.

# 5. SECURITY AND PARAMETER LENGTHS

## 5.1 Attack Scenarios

The fixed passkey is composed of the key $K$ and the check value CV that form the MANA certificate. In the ISO standard [6], only the case where $K$ and CV have equal lengths is considered in the Annex C. This selection of parameters guarantees the required security level in case of one forgery attempt, but the success probability increases as the number of attempts grow. In this section, we identify the possible attacks and determine the security levels as functions of the parameter lengths.

The MANA certificate protocol is protected against a passive wiretapper by the computationally secure components, the Diffie-Hellman key exchange, and the computationally secure check function. The parameter lengths for MANA certificates affect only security against an active Man-in-the-Middle in the Diffie-Hellman protocol. An active attacker has the following two possible ways to run an attack against the legitimate entities at Stage 2 of the Fixed Passkey Association Protocol in Figure 1:

**Attack 1:** A fraudulent device replaces the device's $PK$ in Step 3 by a public key it has generated.

**Attack 2:** A fraudulent host uses its own public key, computes the Diffie-Hellman key base on it and sends the host check in Step 8 based on a guess for the correct $K$.

Clearly, achieving the knowledge of $K$ means a total break of authentication. Success in Attack 2 yields $K$. Success in Attack 1 does not yield $K$ immediately. If the Man-in-the-Middle succeeds in Attack 1, it will know the Diffie-Hellman key computed by the host. Then it only has to wait until it receives the host check $PC_H$ from where it can find $K$ using exhaustive search. Hence both attacks yield total break.

Let $c$ denote the length of $CV$ and $k$ the length of $K$ in decimal digits. We use decimal digits as they are often used in practical implementation and can be directly compared to the check values in the numeric comparison protocol. For more finegrained tuning of the security levels bit lengths can be used.

The purpose of our analysis is to determine how to select $c$ and $k$ to guarantee security of appropriate level, that is, to be comparable with the security that is achieved with the WUSB numeric comparison method. In numeric comparison, the device determines the length of the check value, which is either 2 or 4 digits. Hence the probability that a fraudulent device succeeds in one association attempt is 1/100. On the other hand, the success probability of a fraudulent host to get association with a victim device is 1/10000 if the device requires four-digit check.

In the numeric comparison an association is always conditioned by the user due to the fact that the numeric comparison can only be accepted by a user. In the fixed passkey model, the situation is different and the host and device may be treated differently. On the host's side, the association is always explicitly conditioned by the user by the action of entering the passkey on the host. On the device's side, the association may also be initiated also by a host. This may be the only possibility for devices without proper or accessible user interfaces.

## 5.2 Attack 1 Against Host

Let us now consider Attack 1, where a fraudulent device replaces the device's $PK$ by a $\widehat{PK}$ of its own. It succeeds if $f(\widehat{PK}, K) = f(PK, K)$. It has two possible strategies, where the second can be used repeatedly only if the device remains the same.

1. Select $\widehat{PK}$ at random and send it to the host.
2. Try exhaustively all keys $K$. For each $K$ compute $f(PK, K)$ and generate $\widehat{PK}$ such that $f(\widehat{PK}, K) = f(PK, K)$. Send $\widehat{PK}$ to the host.

If $k > c$ the success probability at each trial according to second strategy is $10^{-c}$ until $10^c$ keys remain to be tested. Hence it is very likely that the success occurs before this. But if $k = c$, the success probability increases at each attempt in strategy 2. Hence if $k > c$, or the host is conditioned to start a new association with different devices, the first strategy is always more powerful.

Attack 1 requires the attacker to have similar capabilities as the Man-in-the-Middle attack against numeric comparison. First, to launch an attack, it must wait the legitimate host and device start an association and then replace the device's protocol message by its own.

We conclude that the success probability of Attack 1 is $10^{-c}$ which is the same as the success probability of the Man-in-the-Middle in the numeric comparison protocol with numeric check of length $c$. But the consequences of success in Attack 1 is the total break which is much worse than if Man-in-the-Middle succeeds in the numeric comparison protocol. However, it is difficult to quantify this difference, the impact of which depends on the application.

If Attack 1 is repeatedly performed against the same device, then each failure can be used to discard all keys $K$ for which $f(\widehat{PK}, K) = f(PK, K)$. The expected number of keys discarded in each failed check is $10^{k-c}$. Let $N_c$ be the number of failed checks allowed in Attack 1. Then after $N_c$ failed checks the expected number of remaining keys $K$ is at least $10^k - N_c 10^{k-c}$. To keep the success probability below $10^{-c}$ we must require that

$$N_c \leq \frac{10^k - 10^c}{10^{k-c}} = 10^c - 10^{2c-k}.$$

If in the numeric comparison model, device can require 4-digit numeric checks, then to achive same security with MANA certificates, a 4-digit $CV$ should be used. Based on the above analysis, the key should be more than 4 digits long, if Attack 1 is launched repeatedly against the same device. This however does not set any essential requirements due to the fact that the requirement imposed by Attack 2, to be considered next, are much more stringent.

## 5.3 Attack 2 Against Device

In Attack 2 a fraudulent host initiates a connection using its own public key and a selected value of $K$ in computing the computationally secure check value in Step 7 and sending it to the device in Step 8. If the device accepts, the attacker has found the correct value of $K$. If $K$ is selected at random, the success probability is $10^{-k}$. If the keys $K$ are tried without repetition against a same device, the expected number of trials the attacker must make to find the correct key is $\frac{1}{2}10^k$. If the attacker is allowed $N_k$ trials then its success probability of success in this attack is $N_k 10^{-k}$.

Compared to Attack 1, this attack is much easier to run against a fixed victim device, in particular, if the device can be activated by the host to start the association.

## 5.4 Attack 1 and Attack 2 Combined

If Attack 1 is repeatedly performed against the same device, then the failure can be used to discard all keys $K$ for which $f(\widehat{PK}, K) = f(PK, K)$. The number of keys discarded in each failed check is $10^{k-c}$. Let $N_c$ be the number of failed checks allowed in Attack 1 as above. Then after $N_c$ failed checks the expected number of remaining keys $K$ is at most $10^k - N_c \cdot 10^{k-c}$. After this the success probability in Attack 2 will be at most

$$N_k \frac{1}{10^k - N_c \cdot 10^{k-c}} = \frac{1}{1 - N_c \cdot 10^{-c}} N_k \cdot 10^{-k}.$$

If $N_c$ is small as it can assumed to be in WUSB context, then the combined attack does not essentially improve the success probability from that of Attack 2 alone.

## 5.5 User Conditioning

As usual in authentication methods based on fixed passkeys, it is necessary to give the user some means to control the number of association attempts that an attacker can make.

The number $N_c$ of trials in Attack 1 against the host is limited in a very strong manner, if to start the association user enters the passkey to the host. Hence in practice, the success probability in Attack 1 will not significantly increase above $10^{-c}$ as long as $k > c$.

To restrict the success probability of Attack 2 to the same level the parameters $c$ and $k$ must be selected to satisfy

$$N_k 10^{-k} \leq 10^{-c},$$

where $N_k$ is the upperbound to the total number of association attempts by fraudulent hosts against a device.

Recall that the total length of the MANA certificate is $k + c$ digits. For example, using 5-digit MANA certificates it is possible to achieve the same security level as with 2-digit numeric comparison. Then it is required to set $N_k = 10$, which may be rather demanding to implement in practice. With 10-digit MANA certificates the security level of 4-digit numeric comparison can be guaranteed if the number $N_k$ of malicious attempts on the device is limited to 100.

The number $N_k$ of adversarial attempts against device can be restiricted by implementing lock-out procedures that require the user to activate the device again. The only way to handle the situation where the upperbound $N_k$ has been achieved is to change the device passkey, that is, generate a fresh $K$ value for the device and a new MANA certificate for the device's public key. The device's public key need not be changed, unless perfect forward secrecy, to be discussed next as a final topic of this security analysis, is a concern.

## 5.6 Perfect Forward Secrecy

In numeric comparison association method device generates a new Diffie-Hellman private exponent and public key for each association. There are two reasons for this, the security of the commitments used in the protocol and the property of perfect forward secrecy. Perfect forward secrecy means that the devices should not contain any secret values that could be used in breaking security of previous associations. In Vaudenay's protocol the commitments are based on independently generated random nonces, but even in this case, perfect forward secrecy would require generating fresh Diffie-Hellman keys for each new association.

Keeping MANA certificate fixed over a multiple of associations is possible only if the device's public key remains the same. Perfect forward secrecy can be added, if so wanted, also to this association method. The idea is to use the shared secret Diffie-Hellman key that is computed based on device's fixed public key only for authentication of a new fresh Diffie-Helman key. It would require the device to generate a fresh Diffie-Hellman key for each association and adding one more authentication check in the protocol.

If WUSB devices typically establish association very rarely and the established connections have long lifetimes, then the benefit of implementing perfect forward secrecy in the association may be small. But for other applications of MANA certificates this aspect of security may be of importance and should be taken into consideration.

# 6. CONCLUSIONS

We investigated how the MANA certificates, which offer a fixed passkey based authentication method for fixed data, could be used for an association method for wireless connections. We demonstrated how the asymmetry of the protocol can be used to implement different security levels for the communicating parties. We presented an application of MANA certificates to the WUSB association models and compared the security properties of the resulting protocol with the security of the existing numeric association protocol. We showed that equivalent security levels can be achieved with sufficiently long, still manageable, passkey values and with proper implementation of user conditioning, time-outs and lock-outs.

## Acknowledgement

The author thanks the members of the WUSB 1.1. working group for useful discussion about different usage scenarios of WUSB hosts and devices.

## 7. REFERENCES

[1] N. Asokan and K. Nyberg. Security associations for wireless devices. In S. Gritzalis, T. Karygiannis, and C. Skianis, editors, *Security and Privacy in Mobile and Wireless Networking*. Troubador Publishing Ltd, Leicester, UK, 2009.

[2] Bluetooth SIG. Bluetooth 2.1 Specification. Bluetooth Special Interest Group `http://www.bluetooth.com/English/Technology/Works/Pages/Core_Specification_v21__EDR.aspx`, 2006.

[3] P. Hunt. Wireless USB Association Models. WUSB Conference, San Jose, May 23-25, 2005. `http://www.usb.org/developers/wusb/docs/presentations/2005/Hunt_-_Wireless_Association_Models.pdf`, 2005.

[4] S. Laur, N. Asokan, and K. Nyberg. Efficient Mutual Data Authentication Using Manually Authenticated Strings. Cryptology ePrint Archive, Report 2005/424, 2005.

[5] S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. In D. Pointcheval, editor, *The 5th International Conference on Cryptology and Network Security, CANS 2006*, volume 4301 of *Lecture Notes in Computer Science*, pages 90–107, Suzhou, China, December 2006. Springer.

[6] Information technology – Security techniques – Entity authentication – Part 6: Mechanisms using manual data transfer . INTERNATIONAL STANDARD ISO/IEC 9798-6, 2005.

[7] J. Suomalainen, J. Valkonen, and N. Asokan. Standards for security associations in personal networks: a comparative analysis. *International Journal of Security and Networks (IJSN)*, 4(1/2):87–100, 2009.

[8] The Transport Layer Security (TLS) Protocol Version 1.2 . RFC 5246 , 2008.

[9] USB Implementers Forum. Wireless USB Specification. Association Models Supplement. Revision 1.0. `http://www.usb.org/developers/wusb/`, 2006.

[10] USB Implementers Forum. Association Models Supplement to the Certified Wireless Universal Bus Specification, Frequently Asked Questions. `http://www.usb.org/developers/wusb/WUSB_AM_FAQ_2007_06_19.pdf`, 2007.

[11] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. Technical Report NRC-TR-2007-002, Nokia Research Center, 2007.

[12] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Proceedings of the Usable Security 2007 (USEC 07) Workshop*, Lowlands, Scarborough, Trinidad/Tobago, February 2007. `http://www.usablesecurity.org/papers/uzun.pdf`.

[13] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 309–326. Springer-Verlag, 2005.

[14] Wi-Fi Alliance. Wi-Fi Protected Setup Specification. Wi-Fi Alliance Document available at `http://www.wi-fi.org/wifi-protected-setup/`, January 2007.

[15] P. R. Zimmermann. Pgpfone: Pretty good privacy phone owner's manual, version 1.0 beta 5, appendix c. `http://web.mit.edu/network/pgpfone/manual/#PGP000057`, January 1996.