DISTANCE-BOUNDING PROOF OF KNOWLEDGE TO AVOID REAL-TIME ATTACKS[12]

Laurent Bussard and Walid Bagga
Institut Eurécom, Corporate Communications, 2229 Route des Crêtes, BP 193,
06904 Sophia Antipolis (France)
{bussard,bagga}@eurecom.fr

Abstract: Traditional authentication is based on proving the knowledge of a private key corresponding to a given public key. In some situations, especially in the context of pervasive computing, it is additionally required to verify the physical proximity of the authenticated party in order to avoid a set of real-time attacks. Brands and Chaum proposed distance-bounding protocols as a way to compute a practical upper bound on the distance between a prover and a verifier during an authentication process. Their protocol prevents frauds where an intruder sits between a legitimate prover and a verifier and succeeds to perform the distance-bounding process. However, frauds where a malicious prover and an intruder collaborate to cheat a verifier have been left as an open issue. In this paper, we provide a solution preventing both types of attacks.

Keywords: Real-time attack, distance-bounding, authentication, proof of knowledge

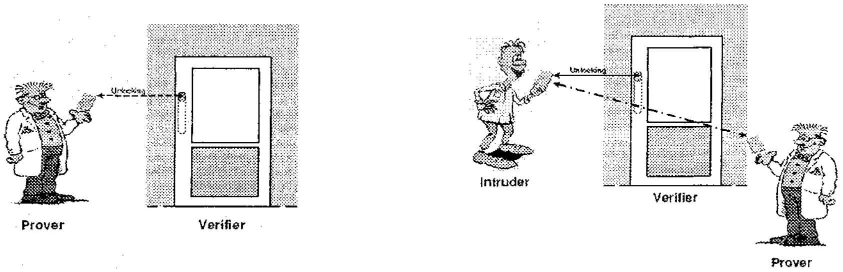
INTRODUCTION

The impressive development in the areas of web technologies, wireless networks, mobile computing, and embedded systems in the past decade has lead to an increasing interest in the topics of pervasive computing and open environments computing. In these contexts, authentication of communicating parties is considered as a major security requirement. As described in [8], a careful authentication may require the verification of the physical proximity of the authenticated party in order to prevent some real-time attacks. A

[12] The work reported in this paper is supported by the IST PRIME project and by Institut Eurécom; however, it represents the view of the authors only.

typical example is applications where digital authentication is required to access a building.

In the scenario depicted in Figure 1-a, a researcher carries around a mobile device (a mobile phone with extended functionalities or a PDA enhanced with communication capabilities) that takes care of computing, storage and communication on his behalf within the laboratory environment. Whenever the researcher approaches the door of a confidential research area, a communication is established between his mobile device and a lock device installed at the door. If the researcher is authorized to access the research area, the door is unlocked. Whenever the combination of the physical proximity and the cryptographic identification is not carefully addressed, some frauds could be performed such as the one depicted in Figure 1-b. In this fraud, a distant researcher (prover) that is allowed to access the confidential research area helps a friend (intruder) that is close by to access the area. For instance, a radio link could be used to establish the communication between the prover and the intruder.



(a) Legitimate Access

(b) Fraudulent Access

Figure 1. Access to a Confidential Research Area

The scenario described above falls under a quite recurring family of security protocols where a prover tries to convince a verifier of some assertion related to his private key. In order to address this problem, Brands and Chaum introduced the concept of distance-bounding protocols in [4]. Such protocols allow determining a practical upper bound on the distance between two communicating entities. This is performed by timing the delay between sending out a challenge bit and receiving back the corresponding response bit where the number of challenge-response interactions is determined by a system-specific security parameter. This approach is feasible if, on one hand, the protocol uses very short messages (one bit) on a dedicated communication channel (e.g. wire, IR) and if, on the other hand, no computation is required during each exchange of challenge-response bits (logical operations on the challenge bit). These conditions allow having round-trip times of few-nanoseconds.

The protocols given in [4] allow preventing *mafia frauds* where an intruder sits between a legitimate prover and a verifier and succeeds to perform the distance-bounding process. In this paper, we provide an extension of such protocols. Our solution allows preventing *terrorist frauds* [8] that have not been addressed so far. In these frauds the prover and the intruder collaborate to cheat the verifier. Note that even if the prover is willing to help the intruder to cheat the verifier, we assume that he never discloses his valuable private key. The key idea in our solution consists of linking the private key of the prover to the bits used during the distance-bounding process. This relies on an adequate combination of the distance-bounding protocol with a bit commitment scheme and a zero-knowledge proof of knowledge protocol [12].

The remainder of this paper is organized as follows. In Section 1, we define the frauds being addressed and give some related work. In Section 2, we draw a general scheme for distance-bounding proof of knowledge protocols, while we give a description of our protocol in Section 3. In Section 4, we analyze the security properties of the proposed protocol. At the end, we conclude and describe further work.

1. PROBLEM STATEMENT

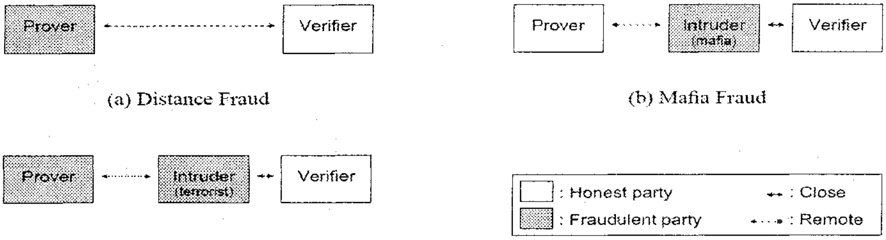
In this section, we provide the definitions of the three attacks we tackle in this paper, namely *distance fraud*, *mafia fraud*, and *terrorist fraud*. Next, we present related work and we show why the existing approaches are not satisfactory.

1.1 Definitions

Distance-bounding protocols have to take into account the three real-time frauds that are depicted in Figure 2. These frauds can be applied in zero-knowledge or minimal disclosure identification schemes. The first fraud is called the *distance fraud* and is defined in the following (Figure 1-a).

DEFINITION 1 (DISTANCE FRAUD) *In the distance fraud two parties are involved, one of them (V the verifier) is not aware of the fraud is going on, the other one (P the fraudulent prover) performs the fraud. The fraud enables P to convince V of a wrong statement related to its physical distance to prover V.*

The distance fraud has been addressed in [4]. This fraud consists of the following: if there's no relationship between the challenge bits and the response bits during the distance-bounding protocol and if the prover *P* is able to know at what times the challenge bits are sent by the verifier *V*, he can make *V* compute a wrong upper bound on his physical distance to *V* by sending out the response bits at the correct time before receiving the challenge bit, regardless of his physical distance to *V*.



(a) Distance Fraud

(b) Mafia Fraud

(c) Terrorist Fraud

Figure 2. Three Real-Time Frauds

The second fraud is called the *mafia fraud* and is defined in the following (Figure 1-b).

Definition 2 (Mafia Fraud) *In the mafia fraud three parties are involved, two of them (P the honest prover and V the verifier) are not aware of the fraud is going on, the third party (I the intruder) performs the fraud. The fraud enables I to convince V of an assertion related to the private key of P.*

The mafia fraud has been first described in [8]. In this fraud, the intruder *I* is usually modeled as a couple $\{\overline{P},\overline{V}\}$ where $\overline{P}$ is a dishonest prover interacting with the honest verifier *V* and where $\overline{V}$ is a dishonest verifier interacting with the honest prover *P*. Thanks to the collaboration of $\overline{V}$, the fraud enables $\overline{P}$ to convince *V* of an assertion related to the private key of *P*. The assertion is that the prover is within a certain physical distance. This fraud was also called *Mig-in-the-middle attack* in [2].

The third fraud is called the *terrorist fraud* and is defined in the following (Figure 1-c).

Definition 3 (Terrorist Fraud) *In the terrorist fraud three parties are involved, one of them (V the verifier) is not aware of the fraud is going on, the two others (P the dishonest prover and I the intruder or terrorist) collaborate to perform the fraud. Thanks to the help of P, the fraud enables I to convince V of an assertion related to the private key of P.*

The terrorist fraud has been first described in [8]. In this fraud, the prover and the intruder collaborate to perform the fraud whereas in the mafia fraud the intruder is the only entity that performs the fraud. Note that the prevention of terrorist frauds implies the prevention of mafia frauds.

1.2 Related Work

In this section we review different techniques that have been proposed and show why they are not sufficient when it is necessary to verify that some entity knowing a private key is indeed physically present.

Constrained Channel [13] aims at exchanging some secret between two physical entities and thus ensures the proximity of two devices. An obvious implementation is to have a physical contact [16] between the two artifacts. This scheme works only when the attacker is not physically present. It can protect a system only against distance frauds.

Context Sharing is a straightforward extension of constrained channels where some contextual data is used to initiate the key exchange. For instance, in [10], the pairing mechanism is done by shaking artifacts together in order to create a common movement pattern that is subsequently used to bootstrap the security of communications. This approach prevents distance frauds and can partially avoid mafia frauds when the context is difficult to reproduce.

Isolation [3] is a widely deployed solution to check whether a physical entity holds a secret. The device is isolated in a Faraday cage during a challenge-response protocol. This solution prevents distance frauds, mafia frauds as well as terrorist frauds. However, it is difficult to deploy, it is not user-friendly, and does not allow mutual authentication.

Unforgeable Channel aims at using communication channels that are difficult to record and reconstruct without knowing some secret. For instance, channel hopping [1] or radio frequency watermarking [11] makes it difficult to transfer data necessary to create the signal in another place. This scheme protects against distance frauds and the solution proposed in [1] can prevent mafia frauds as well when it is not possible to identify communication sources. Quantum cryptography can also be envisioned as an unforgeable channel.

Time of Flight relies on the speed of sound and/or light. Sound and especially ultra-sound [15] is interesting to measure distance since it is slow enough to authorize computation without reducing the accuracy of the measure. Sound-based approaches cannot protect against physically present attackers and thus can only prevent distance frauds. Some works also rely on the speed of light when measuring the round trip time of a message to evaluate the distance to the prover. However, one meter accuracy implies responding within few nanoseconds and thus it cannot be done through standard communication channels and cannot use cryptography [17]. Such schemes prevent distance frauds and the solution proposed in [4] prevents mafia frauds as well.

As shown above, only *isolation* allows preventing distance, mafia and terrorist frauds all together. In this paper, we focus on *distance-bounding* *protocols* and propose a solution that prevents the three real-time attacks. In contrast with *isolation*, our approach is easy to deploy and allows mutual authentication.

2. THE GENERAL SCHEME

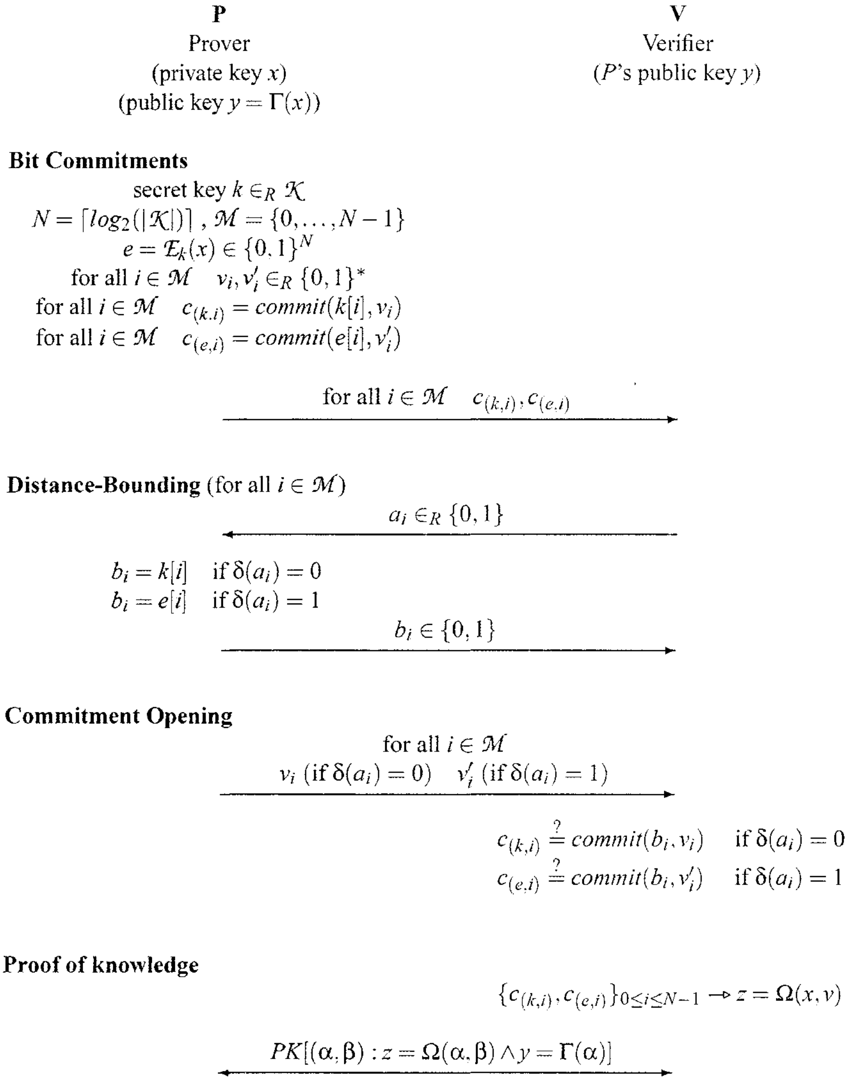
In this section, we present a general scheme (denoted **DBPK**) containing the basic building blocks of *distance-bounding proof of knowledge* protocols.

2.1 Description

The **DBPK** scheme is depicted in Table 1. It relies on a set of global settings that have to be performed before the execution of any interaction between the prover and the verifier. Besides the cryptosystem's public parameters, these global settings allow the prover to have a valuable private key and a certificate on the corresponding public key. That is, before any interaction with the verifier, the prover holds a private key x which importance, by assumption, is so high that the prover should not reveal it to any other party. In addition, the prover holds a certificate (generated by a globally trusted authority) on its public key $y = \Gamma(x)$.

The first stage of the **DBPK** protocol is called the *Bit Commitment* stage. During this stage the prover first picks a random one-time key $k \in_R \mathrm{K}$ and uses it to encrypt its private key x according to a publicly known symmetric key encryption algorithm E. This leads to the ciphertext $e = \mathrm{E}_k(x)$. Once the encryption performed, the prover commits to each bit of both k and e according to a secure bit commitment scheme *commit*. For each bit $k[i]$ (resp. $e[i]$), a string v_i (resp.v'_i) is randomly chosen by the prover to construct the commitment blob $c_{(k,i)}$ (resp. $c_{(e,i)}$).

Once the *Bit Commitments* stage is completed, the actual distance-bounding interactions are executed during the *Distance-Bounding* stage. Basically, N interactions are performed between the prover and the verifier. In the i^{th} interaction, the prover releases either $k[i]$ or $e[i]$ depending on whether the challenge bit is equal to 0 or to 1. Note that $k[i]$ (resp. $e[i]$) denotes the i^{th} bit in the binary representation of k (resp. e) where $k[0]$ (resp. $e[0]$) is the least significant bit of k (resp. e). During each bit exchange, the round trip time (few nanoseconds) is measured in order to verify the distance to the prover.



P
Prover
(private key x)
(public key $y = \Gamma(x)$)

V
Verifier
(P's public key y)

Bit Commitments

secret key $k \in_R \mathcal{K}$
$N = \lceil log_2(|\mathcal{K}|) \rceil$, $\mathcal{M} = \{0, \ldots, N-1\}$
$e = \mathcal{E}_k(x) \in \{0,1\}^N$
for all $i \in \mathcal{M}$ $\quad v_i, v'_i \in_R \{0,1\}^*$
for all $i \in \mathcal{M}$ $\quad c_{(k,i)} = commit(k[i], v_i)$
for all $i \in \mathcal{M}$ $\quad c_{(e,i)} = commit(e[i], v'_i)$

$$\xrightarrow{\text{for all } i \in \mathcal{M} \quad c_{(k,i)}, c_{(e,i)}}$$

Distance-Bounding (for all $i \in \mathcal{M}$)

$$\xleftarrow{a_i \in_R \{0,1\}}$$

$b_i = k[i]$ if $\delta(a_i) = 0$
$b_i = e[i]$ if $\delta(a_i) = 1$

$$\xrightarrow{b_i \in \{0,1\}}$$

Commitment Opening

$$\xrightarrow[\text{}]{\text{for all } i \in \mathcal{M} \quad v_i \text{ (if } \delta(a_i) = 0) \quad v'_i \text{ (if } \delta(a_i) = 1)}$$

$c_{(k,i)} \overset{?}{=} commit(b_i, v_i)$ if $\delta(a_i) = 0$
$c_{(e,i)} \overset{?}{=} commit(b_i, v'_i)$ if $\delta(a_i) = 1$

Proof of knowledge

$\{c_{(k,i)}, c_{(e,i)}\}_{0 \le i \le N-1} \rightarrow z = \Omega(x, v)$

$$\xleftrightarrow{PK[(\alpha, \beta) : z = \Omega(\alpha, \beta) \wedge y = \Gamma(\alpha)]}$$

Figure 3. A general scheme for $DBPK[\alpha : y = \Gamma(\alpha)]$

After the execution of the N successful challenge-response bit exchanges, the prover opens the commitments on the released bits of k and e. The *Commitment Opening* stage consists of sending the string v_i if $k[i]$ has been released and v'_i otherwise. Only half of the bits of k and e are released to the verifier. This must not allow the verifier to get any significant information about the valuable private key x. In the case where the verification of $c_{(k,i)}$ (resp. $c_{(e,i)}$) fails, the verifier sends back an error notification of the form $error_k(i)$ (resp. $error_e(i)$).

The last step in the **DBPK** protocol is the *Proof of Knowledge* stage. During this stage, the prover convinces the verifier in a zero-knowledge interaction that he is the party who performed the three previously described stages. That is, the prover proves that he has generated the different commitments, that the generated commitments correspond to a unique private key, and that this private key corresponds to the public key y that is used by the verifier to authenticate the prover. Before the proof of knowledge process can be performed, the verifier must compute a one way function on the private key x: $z = \Omega(x,y)$ where v and x are known only by the prover. As z depends on and only on the commitments on the bits of k and e, it may even be computed just after the *Bit Commitments* stage. The proof of knowledge we use is denoted $PK[(\alpha,\beta) : z = \Omega(\alpha,\beta) \wedge y = \Gamma(\alpha)]$ where the Greek small letters denote the quantity the knowledge of which is being proved, while all other parameters are known to the verifier. The functions Ω, Γ, δ, E, and *commit* are adequately chosen to meet our security requirements, namely the prevention of the distance, mafia, and terrorist frauds.

The distance-bounding proof of knowledge of a secret x such that $y=\Gamma(x)$ is denoted $DBPK[\alpha : y = \Gamma(\alpha)]$.

3. OUR PROTOCOL

This section presents a concrete distance-bounding proof of knowledge protocol that consists of exactly the same building blocks of the **DBPK** protocol. The protocol will be denoted **DBPK-Log** = $DBPK[\alpha : y = g^{\alpha}]$.

3.1 Global Settings

We first describe the global settings on which relies the **DBPK-Log** protocol. These settings consist of two main phases: *Initialization* and *Registration*. In the *Initialization* stage, a trust authority (TA) provides the public parameters of the system.

Initialization:

TA sets up the system's global parameters

- TA chooses a large enough strong prime p, i.e. there exists a large enough prime q such that $p = 2q+1$
- TA chooses a generator g of Z_p^*
- TA chooses an element $h \in_R Z_p^*$

The randomly chosen element h will be used by the commitment scheme. The only requirement is that neither of the prover and the verifier knows $\log_g(h)$. This can be achieved either by letting the trusted authority generate this element, or by making the prover and the verifier jointly generate h. The two alternatives rely on the intractability of the *discrete logarithm* problem [14].

In the *Registration* stage, a user chooses a private key and registers at the trust authority so to get a certificate on the corresponding public key.

Registration:
The following steps are taken by P to get a certified public key corresponding to a valuable private key

- P selects an odd secret $x \in_R Z_{p-1} \setminus \{q\}$, then computes $y = g^x$. The public key of P is y and his private key is x
- P registers his public key with TA so TA publishes a certificate on this public key

Note that the two phases described above are executed only once. They allow generating the prover's public and private keys that will be used in the different subsequent distance-bounding proofs of knowledge.

3.2 Interactions

Our distance-bounding proof of knowledge protocol starts with the *Bit Commitments* stage where the prover P generates a random key k, and uses this key to encrypt the private key x. Then, P performs a secure commitment on each bit of the key k and encryption e.

Bit Commitments:
The following steps are performed

- Given a security parameter m', P picks at random $k \in_R \{0, \ldots, 2^N-1\}$, where $N = m'+m$ and $m = \lceil log_2(p) \rceil$.
- P computes $e \in \{0,\ldots, 2^N-1\}$ such that $e \equiv x-k \bmod p-1$.
- For all $i \in \{0, \ldots, N-1\}$, P chooses $v_{k,i}$, $v_{e,i} \in_R Z_{p-1}$, computes $c_{k,i} = g^{k[i]} \cdot h^{v_{k,i}} \bmod p$ and $c_{e,i} = g^{e[i]} \cdot h^{v_{e,i}} \bmod p$, then sends $c_{k,i}$ as well as $c_{e,i}$ to V

Once the verifier V receives all the commitment blobs corresponding to the bits of k and e, the *Distance-Bounding* stage can start. Thus, a set of fast single bit challenge-response interactions is performed. A challenge corresponds to a bit chosen randomly by V while a response corresponds either to a bit of k or to a bit of e.

Distance-Bounding:
For all $i \in \{0, \ldots, N\text{-}1\}$,

- V sends a challenge bit $a_i \in_R \{0,1\}$ to P
- P immediately sends the response bit $b_i = \overline{a}_i k[i] + a_i e[i]$ to V

At the end of the *Distance-Bounding* stage, the verifier V is able to compute an upper bound on the distance to P. In order to be sure that P holds the secrets k and e, the prover P opens, during the *Commitment Opening* stage, the commitments on the bits of k and e that have been released during the *Distance-Bounding* stage.

Commitment Opening:
The commitments of the released bits are opened. If all the checks hold, all the bit commitments on k and e are accepted, otherwise they are rejected and an error message is sent back

- For all $i \in \{0, \ldots, N\text{-}1\}$, P sends $\overline{a}_i v_{k,i} + a_i v_{e,i}$ to V
- For all $i \in \{0, \ldots, N\text{-}1\}$, V performs the following verification:

$$\overline{a}_i c_{k,i} + a_i c_{e,i} \underset{?}{=} g^{\overline{a}_i k[i] + a_i e[i]} \cdot h^{\overline{a}_i v_{k,i} + a_i v_{e,i}} \bmod p$$

The proof of knowledge allows the verifier V to be sure that e is indeed the encryption of the private key x corresponding to the public key y of the prover. From the bit commitments, V can compute:

$$\begin{aligned} z &= \prod_{i=0}^{N-1} (c_{k,i} \cdot c_{e,i})^{2^i} = g^{\sum_{i=0}^{N-1} (2^i \cdot k[i] + 2^i \cdot e[i])} \cdot h^{\sum_{i=0}^{N-1} (2^i \cdot (v_{k,i} + v_{e,i}))} \\ &= g^{k+e} \cdot h^{v} = g^{x} \cdot h^{v} \quad \bmod p \end{aligned}$$

Note that V is able to compute z as soon as all the commitments on the bits of k and e are received.

Proof of Knowledge:
Given $z = g^x \cdot h^v$, the following proof of knowledge is performed by P and V:
$PK[(\alpha,\beta) : z = g^{\alpha} h^{\beta} \wedge y = g^{\alpha}]$.

4. SECURITY ANALYSIS

In this section, we discuss the relevant security properties of the **DBPK-Log** protocol. First, we show that our protocol prevents distance, mafia, and terrorist frauds. Next, the security properties of the encryption scheme that is used to hide the prover's private key are studied.

4.1 Preventing Distance, Mafia and Terrorist Frauds

The first security requirement for our distance-bounding proof of knowledge protocol is a correct computation of an upper bound on the distance between the prover and the verifier. This requirement is being already achieved in the **DBPK** general scheme according to the following.

Proposition 4.1 *If the **DBPK** protocol is performed correctly, then the distance fraud has a negligible probability of success.*

Proof: Assume that the prover P knows at what times the verifier V will send out bit challenges. In this case, he can convince V of being close by sending out the bit response b_i at the correct time before he receives the bit a_i. The probability that P sends correct responses to V before receiving the challenges is equal to

$$\Pi_{i=1}^{N}(P[b_i = k[i]|\delta(a_i) = 0] + P[b_i = e[i]|\delta(a_i) = 1]) = 2^{-N}$$

In Proposition 4.1, the correct execution of the protocol means that each party performs exactly and correctly the actions specified in the different steps of the protocol.

The **DBPK-Log** protocol is an implementation of the **DBPK** protocol where the function δ corresponds to the identity function, i.e. $\forall i\ \delta(a_i)= a_i$. This leads to the following proposition.

Proposition 4.2 *If the **DBPK** protocol is performed correctly, then the distance fraud has a negligible probability of success.*

Respecting the notations of Section 2, we introduce the three following properties.

Property 4.1 *Let* $\Gamma : \{0,1\}^* \to \{0,1\}^*$ *be the function such that* $y = \Gamma(x)$, *then the following holds:*

- *Given* y, *it is hard to find* x *such that* $y = \Gamma(x)$.
- *It is hard to find* $x \neq x'$ *such that* $\Gamma(x) = \Gamma(x')$.

Property 4.2 *Let* $\Omega : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ *be the function such that* $z = \Omega(x,v)$, *then the following holds*

- *Knowing* z *and* Ω, *it is hard to find* (x,v).
- *It is hard to find* $(x,v) \neq (x',v')$ *such that* $\Omega(x,v) = \Omega(x',v')$.

Property 4.3 *Let* E *be the function such that* $e = E_k(x)$, *then the following holds*

(a) E *is an encryption scheme: knowing* e *and* E, *it is hard to find* x *without knowing* k; *and given* e *and* k, $x = D_k(e)$ *is efficiently computable.*

(b) Given either k[i] or e[i] for all i ∈ {0,...,N-1}, it is hard to find x.
(c) It is efficient to compute z = Ω(x,v) from the commitments on the bits of k and e.

The second security requirement for distance-bounding proof of knowledge protocols consists in preventing terrorist frauds. This requirement can already be achieved in the **DBPK** general scheme according to the following proposition.

Proposition 4.3 *If Property 4.1, Property 4.2, and Property 4.3 are respected and if the **DBPK** protocol is performed correctly, then the terrorist fraud has a negligible probability of success.*

Proof: A successful execution of the Proof of Knowledge stage proves that the entity knowing the private key corresponding to the public key y have performed the *Bit Commitments* stage. Assume that the latter has been performed using *k* and *e*. Then, the probability for an intruder to perform the Distance-Bounding stage successfully using (k', e') ≠(k,e) is equal to

$$\Pi_{i=1}^{N}(P[k[i] = k'[i]|\delta(a_i) = 0] + P[e[i] = c'[i]|\delta(a_i) = 1]) = 2^{-N}$$

This shows that without knowing (k,e), i.e. without knowing $x = D_k(e)$, the probability of success of a terrorist fraud is negligible.

The **DBPK-Log** consists of the same building blocks than those of the **DBPK** protocol. Moreover, the three following statements hold

(1) The function Γ : x → g^x respects Property 4.1 thanks to the intractability of the discrete logarithm problem.
(2) The function Ω : (x,v) → $g^x \cdot h^v$ respects Property 4.2 thanks to the intractability of the *representation* problem.
(3) The one-time pad $E_k(x)$ = x-k mod p-1 respects Property 4.3 (see Section 4.2).

The properties listed above lead to the following.

Proposition 4.4 *If the **DBPK-Log** protocol is performed correctly, then the terrorist fraud has a negligible probability of success.*

Recall that the prevention of terrorist frauds makes the prevention of mafia frauds straightforward.

4.2 Encryption of the Private Key

Since some bits of the key *k* are revealed, it is straightforward that a one-time key is necessary. To be compliant with Property 4.3, we propose a dedicated one-time pad: $e = E_k(x) = x-k \bmod p-1$ where $k \in \{0,\dots, 2^N-1\}$ is randomly chosen before each encryption. The parameter N is such that $N = m+m'$, where m is the number of bits of the private key x and m' is a system-specific security parameter. The prime number p is a strong prime, i.e. $p = 2q+1$ where q is an enough large prime. This scheme is compliant with the following:

- Property 4.3.a: with this encryption scheme, revealing e still ensures perfect secrecy of x: $P_{X|E}(X = x| E = e) = P_X(X = x) = 2^{-N}$ for all x,e
- Property 4.3.b: in the following, we show that the knowledge of either $k[i]$ or $e[i]$ for all $i \in \{0,\dots,N-1\}$, allows to retrieve information on x with probability less than $2^{-2m'}$. We basically study the impact of the security parameter m' on the probability of revealing information on x. Knowing b such that $b = x-b'$, information on x can be statistically obtained when a large enough number n of samples can be collected to have a sample mean $\overline{Y}_n$ close to the mean μ i.e. $|\overline{Y}_n-\mu| < p-1$. The *Central Limit Theorem* states that the sum of a large number of independent random variables has a distribution that is approximately normal. Let $Y_1,Y_2,\dots, Y_n$ be a sequence of independent and identically distributed random variables with mean μ and variance σ^2 and $\overline{Y}_n = (1/n)\sum_{i=1}^{n} Y_i$ then,

$$\frac{\sqrt{n}(\overline{Y}_n - \mu)}{\sigma} \to N(0,1), \text{ when } n\to\infty$$

That is

$$P\left\{-a \le \frac{\sqrt{n}(\overline{Y}_n - \mu)}{\sigma} \le a\right\} \to \frac{1}{\sqrt{2\pi}}\int_{-a}^{a} e^{-z^2/2}dz$$

Since the following holds

$$P\left\{-a \le \frac{\sqrt{n}(\overline{Y}_n - \mu)}{\sigma} \le a\right\} = P\left\{\mu - a\frac{\sigma}{\sqrt{n}} \le \overline{Y}_n \le \mu + a\frac{\sigma}{\sqrt{n}}\right\}$$

then, the probability of having a sample mean close to the mean is

$$P\{\mu - 2^{m-1} \le \overline{Y}_n \le \mu + 2^{m-1}\} \to \frac{1}{\sqrt{2\pi}}\int_{-2^{m-1}\frac{\sqrt{n}}{\sigma}}^{2^{m-1}\frac{\sqrt{n}}{\sigma}} e^{-z^2/2}dz$$

The mean is $\mu=x+2^{N-1}$ and the variance is σ^2, where $\sigma= (2^N)^2/12$. Hence, $(2^{m-1}\sqrt{n})/\sigma = (6\sqrt{n})/2^{m+2m'} >> 1$. In other words,

$$P\{\mu - 2^{m-1} \le \overline{Y}_n \le \mu + 2^{m-1}\} \cong \frac{12\sqrt{n}}{2^{m+2m'}}\cdot\frac{1}{\sqrt{2\pi}}$$

We study the number of samples necessary to get information on x:

$$\frac{12\sqrt{n_0}}{2^m} \cdot \frac{1}{\sqrt{2\pi}} = \frac{12\sqrt{n}}{2^{m+2m'}} \cdot \frac{1}{\sqrt{2\pi}} \Rightarrow \frac{n}{n_0} = 2^{4m'}$$

Here n_0 is the number of sample necessary when $m' = 0$, i.e. when the parameters x, k, and e are of equal length. In the worst case, n_0 is equal to 1 and the number of samples necessary to get information on x is $2^{4m'}$.

Note that the security parameter *m*0 ensures a probability of successful attack less than $2^{-4m'}$ at the expense of m' additional challenge-response bit exchanges. For instance, for $m = 1024$ bits and $m' = 50$ bits, the probability of retrieving information about x is less than 2^{-200} at the expense of around 5% of additional challenge-response bit exchanges.

- Property 4.3.c: it is possible to deduce a representation of z depending on x from commitments on bits of k and e (Section 3):
  $z = \prod_{i=0}^{N-1} (c_{k,i} \cdot c_{e,i})^{2^i} = g^x \cdot h^v \bmod p$

CONCLUSION AND FURTHER WORK

In this paper, we addressed the problem of terrorist frauds in application scenarios where cryptographic authentication requires the physical proximity of the prover. Our solution consists in distance-bounding proof of knowledge protocols that extend Brands and Chaum's distance-bounding protocols [4]. We first presented a general scheme that shows the main building blocks of such protocols. We then presented a possible implementation of such protocols and analyzed its security properties. Even though we have not reached perfect secrecy, our solution remains secure in the statistical zero-knowledge security model.

The general scheme presented in this paper (**DBPK**) could be used with any public key scheme Γ if adequate commitment scheme *commit*, encryption method E, and representation function Ω exist. We proposed a solution relying on a public key scheme based on the discrete logarithm problem, bit commitment based on discrete logarithm, group addition one-time pad, and representation problem: **DBPK-Log** = $DBPK[\alpha : y = g^{\alpha}]$. This scheme could directly be used with ElGamal's and Schnorr's identification schemes that both rely on the discrete logarithm problem.

The integration of distance-bounding with Fiat-Shamir identification scheme [9] is not straightforward. The public key x is chosen in Z_n where $n = pq$ and the public key is *x*2 mod n. It is necessary to define $DBPK[\alpha : y = \alpha^2]$. Using the commitment scheme presented in this paper, the following proof of knowledge is required: $PK[\alpha,\beta : z = g^{\alpha} \cdot h^{\beta} \wedge y = \alpha^{2}]$. In other words, the parameter g has to be a generator of a cyclic group of order n.

We are also studying whether such a scheme can be used in a privacy preserving way. **DBPK** could be integrated in a group signature scheme, e.g. the initial one proposed in [7] would be: $DBPK[\alpha : \tilde{z} = \tilde{g}^{(a^{\alpha})}]$; $PK[\beta : \tilde{z}\tilde{g} = \tilde{g}^{(\beta^{e})}]$.This way, the verifier can verify that he is in front of a member of some group. However the verifier does not get any information on the identity of this group member. In this case, the encryption has to be done modulo n. A further step would be the integration of distance-bounding protocols in unlinkable and/or pseudonymous credentials schemes such as Idemix [6].

An alternative way to address terrorist frauds would be by combining trusted hardware with any protocol preventing mafia frauds [5]. In other words, a tamper-resistant hardware trusted by the verifier has to be used by the prover to execute the protocol. However, our approach is more general and easier to deploy since it neither relies on tamper-resistant hardware nor requires device certification process.

REFERENCES

[1] A. Alkassar and C. Stuble. Towards secure iff: preventing mafia fraud attacks. In *Proceedings of MILCOM 2002*, volume 2, pages 1139–1144, October 2002.

[2] Ross Anderson. *Security Engineering: A Guide to Building Dependable distributed Systems*. John Wiley and Sons, 2001.

[3] S. Bengio, G. Brassard, Y. Desmedt, C. Goutier, and J.J. Quisquater. Secure implementation of identification systems. *Journal of Cryptology*, 4(3):175–183, 1991.

[4] S. Brands and D. Chaum. Distance-bounding protocols (extended abstract). In *Proceedings of EUROCRYPT 93*, volume 765 of *LNCS*, pages 23–27. Springer-Verlag, May 1993.

[5] L. Bussard and Y. Roudier. Embedding distance-bounding protocols within intuitive interactions. In *Proceedings of Conference on Security in Pervasive Computing (SPC'2003)*, LNCS. Springer, 2003.

[6] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *Lecture Notes in Computer Science*, 2045, 2001.

[7] J. L. Camenisch and M. A. Stadler. Efficient group signature schemes for large groups. In *Advances in Cryptology – CRYPTO '97 Proceedings*, volume 1294 of *LNCS*, pages 410–424. Springer-Verlag, 1997.

[8] Yvo Desmedt. Major security problems with the 'unforgeable' (Feige)-Fiat-Shamir proofs of identity and how to overcome them. In *Proceedings of SecuriCom '88*, 1988.

[9] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology—Crypto '86*, pages 186–194, New York, 1987. Springer-Verlag.

[10] L.E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H-W. Gellersen. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Proceedings of UbiComp 2001*, 2001.

[11] Yih-Chun Hu, A. Perrig, and D.B. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *Proceedings of INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1976–1986, March 2003.

[12] J.Pieprzyk, T.Hardjono, and J.Seberry. *Fundamentals of Computer Security.* Springer, 2003.

[13] T. Kindberg, K. Zhang, and N. Shankar. Context authentication using constrained channels. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 14–21, June 2002.

[14] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. *Handbook of Applied Cryptography.* CRC Press, Inc., 1996.

[15] N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *Proceedings of the 2003 ACM workshop on Wireless security*, 2003.

[16] Frank Stajano and Ross J. Anderson. The resurrecting duckling: Security issues for adhoc wireless networks. In *Security Protocols Workshop*, pages 172–194, 1999.

[17] B. Waters and E. Felten. Proving the location of tamper-resistant devices. Technical report.