Institut EURECOM
Corporate Communications
2229, route des Crêtes BP 193
06904 Sophia Antipolis
(France)

Research Report[a] N^o 109 — RR-04-109

Distance-Bounding Proof of Knowledge Protocols to Avoid Terrorist Fraud Attacks

Laurent Bussard and Walid Bagga

May 14, 2004

[a]Eurecoms research is partially supported by its industrial partners: Bouygues Telecom, Fondation d'entreprise Groupe Cegetel, Fondation Hasler, France Telecom, Hitachi, ST Microelectronics, Swisscom, Texas Instruments, and Thales

Distance-Bounding Proof of Knowledge Protocols to Avoid Terrorist Fraud Attacks

Laurent Bussard and Walid Bagga

Institut Eurécom
Corporate Communications
2229, route des Crêtes BP 193
06904 Sophia Antipolis (France)

May 14, 2004

Abstract

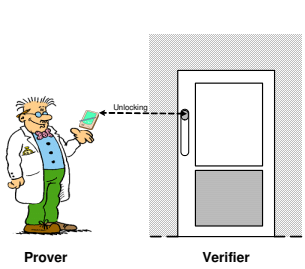
Real-time frauds can be applied against numerous zero-knowledge or minimal disclosure identification schemes that protect physical services, be it opening a door or verifying attributes of a certified device. In [4], Brands and Chaum proposed distance-bounding protocols to forbid mafia fraud attacks and let the terrorist fraud attack as an open issue. In this paper, we describe an extension of the initial scheme in order to forbid both mafia and terrorist fraud attacks.

Keywords: terrorist fraud, mafia fraud, distance-bounding, proof of knowledge

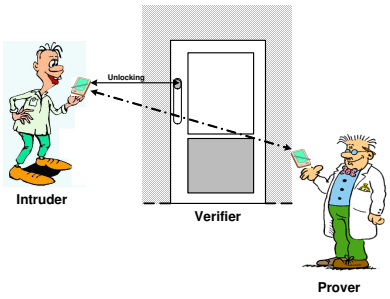
1 Introduction

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 topics, the trustworthiness of the communicating entities should be carefully addressed. In this paper, we focus on applications combining physical proximity and cryptographic identification schemes. Typical ones are those where digital identification is required to access a building or, more generally, to enter in a given area. Our research is applicable to other application areas such as those described in[6, 9, 17, 1, 13].

Figure 1(a) shows a possible real-world application scenario. In this scenario, 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 in the research 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 device lock 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. A radio link could be used, for instance, 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 applications in cryptography where a prover tries to convince a verifier of some assertion related to his private key. The assertion in our case is that the prover is within a certain physical distance. Brands and Chaum [4] were the first to specifically address this problem. They introduced *distance-bounding* protocols that allow to determine 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. The number of challenge-response interactions being determined by a defined security parameter. This approach is feasible if and only if the protocol uses very short messages (one bit) on a dedicated communi-

cation channel (e.g. wire, IR) and if no computation is required during each challenge-response exchange (few logical operations). This allows having round-trip times of few-nanoseconds.

The protocols given in [4] allow to prevent the 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 those protocols. Our solution allows preventing terrorist frauds that have not been addressed so far. In these frauds the prover and the intruder collaborate to cheat the verifier. However, even if the prover helps the intruders, the prover does not want to reveal his valuable private key.

The remainder of this paper is organized as follows. Section 2 defines the frauds being addressed and gives some related work. Section 3 presents a general scheme for distance-bounding proof of knowledge protocols and gives the main security requirements. Section 4 contains a description of our protocol which security analysis is given in Section 5. At the end, we conclude and describe further work.

2 Problem Statement

In this section, we define 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 satifactory.

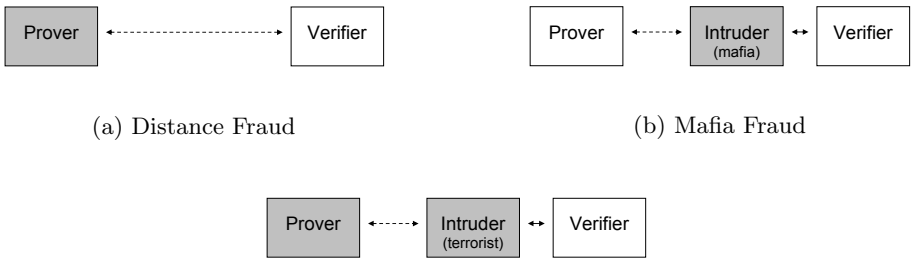
2.1 Definitions

Distance-bounding protocols have to take into account the three real-time frauds that are depicted in Figure 2. Those frauds are not called *man-in-the-middle attacks* because the intruder does not deal with security protocol but transparently forwards challenges and responses.

The first fraud is called the *distance fraud* and is defined in the following (Figure 2(a)).

Definition 1 (Distance Fraud). *A* distance fraud *is a real-time fraud that can be applied in location schemes. In the 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 V.*

The distance fraud has been discussed in [4]. This fraud consists on the following: if there's no relationship between the challenge bits and the re-



(a) Distance Fraud

(b) Mafia Fraud

(c) Terrorist Fraud

Figure 2: Three Real-Time Frauds

sponse bits during the distance-bounding protocol and if the prover P knows at what times the challenge bits are sent by the verifier V, he can make V compute a wrong upper-bound of 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.

The second fraud is called the *mafia fraud* and is defined in the following (Figure 2(b)).

Definition 2 (Mafia Fraud). *A* mafia fraud *is a real-time fraud that can be applied in zero-knowledge or minimal disclosure identification schemes by an intruder I. In the 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 a statement related to the private key of P.*

The mafia fraud has been first described by Desmedt in [9]. In this fraud, the intruder I is usually modeled as a couple $\{\bar{P}, \bar{V}\}$ where $\bar{P}$ is a dishonest prover interacting with the honest verifier V and where $\bar{V}$ is a dishonest verifier interacting the dishonest prover P. Thanks to the collaboration of $\bar{V}$, the fraud enables $\bar{P}$ to convince V of a statement related to the private key of P. 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 2(c)).

Definition 3 (Terrorist Fraud). *A* terrorist fraud *is a real-time fraud that can be applied in zero-knowledge or minimal disclosure identification*

schemes by fraudulent prover P and Intruder I cooperating together. In the 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 a statement related to the private key of P.

The terrorist fraud has been first described in [9]. 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.

2.2 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 someone is indeed physically present:

- *Constrained Channels* [14] aim at exchanging some secret between two physical entities and thus assure the proximity of two devices. An obvious implementation is to have a physical contact [17] between both artifacts.

  This scheme only works when the attacker is not physically present. It can only protect a system 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 forbids 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 approach is used in ATMs to check whether a smart card embeds a private key corresponding to a given public key.

  This solution forbids distance frauds, mafia frauds, as well as terrorist frauds. However, it is difficult to deploy, it is not userfriendly, and does not allow mutual authentication.

- *Unforgeable Channel* aims at using communication media that is difficult to forward without knowing some secret. For instance, channel hopping [1] or RF watermarking [11] makes it difficult to transfer data necessary to create the signal in another place.

  This scheme protects against distance frauds and [1] protects against mafia frauds when it is not possible to identify communication sources.

- *Time of Flight* relies on the speed of sound and/or light. Sound and especially ultra-sound [16] is interesting to measure distance because 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 only forbid 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 implies cryptography [18].

  Protection against distance fraud and even mafia fraud [4].

2.3 Our Contribution

Only *isolation* and *distance bounding protocols* are sure and precise enough to defeat an intruder who is in front of the verifier (i.e. mafia and terrorist frauds). Isolation is not user-friendly enough and cannot ensure mutual authentication. Thus this paper focuses on an extension of the distance bounding protocol proposed by Chaum in [4]. In our solution, we keep the same constraints: one bit exchanges and no cryptography during the rapid bit exchange.

We introduce *distance-bounding proof of knowledge* protocols that are adequate combinations of distance-bounding protocols [4], bit commitment schemes and zero-knowledge proof of knowledge protocols [12]. Our construction allows preventing the three frauds described above.

3 The General Scheme

In this section, we present a general scheme that is considered as the basis for *distance-bounding proof of knowledge* protocols. The described scheme will be denoted **DBPK**.

3.1 Description

The **DBPK** protocol is depicted in Table 1. This scheme 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 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 key $k \in_R \{0,1\}^*$ and uses it to encrypt its private key x according to a publicly known symmetric key encryption method $\mathcal{E} : \{0,1\}^* \rightarrow \{0,1\}^*$. This leads to $e = \mathcal{E}_k(x) \in \{0,1\}^*$. Note that as in every encryption scheme, the knowledge and only the knowledge of both e and k allows to compute the private key $x = \mathcal{D}_k(e)$. 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, m 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).

After the execution of the m challenge-response bit exchanges, the verifier asks the prover to open the commitments on the released bits of k and e. The *Commitment Opening* stage consists on sending the string v_i if $k[i]$ has been released and $v'[i]$ otherwise. Note that only some of the bits of k and e are released to the verifier. This should not allow the verifier or any other

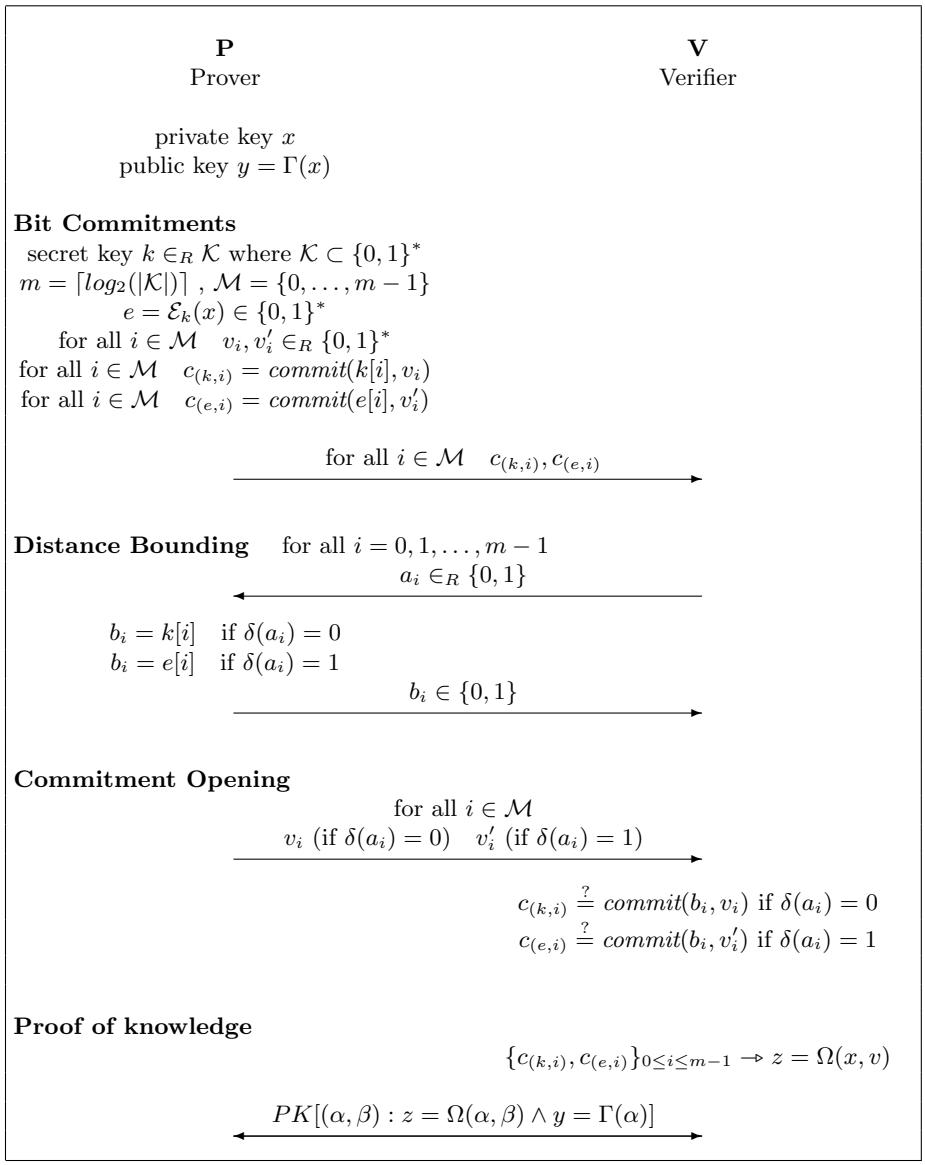


Table 1: A general scheme for $DBPK[\alpha : y = \Gamma(\alpha)]$

party sniffing the interactions between the prover and 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. Note that before the proof of knowledge process can be performed, the verifier must compute $z = \Omega(x, v)$ where v is 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 letters denote the quantity the knowledge of which is being proved, while all other parameters are known to the verifier.

To sum up, we point out, in the following, the major principles behind the general scheme described above. We define by $DBPK[\alpha : y = \Gamma(\alpha)]$ the distance-bounding proof of knowledge of a secret x so that $y = \Gamma(x)$. The principle of the scheme can be sketched as follows:

(1) Distance bounding phase implies that someone knowing k' and e' is close.

(2) Bit commitments on bits of k and e can be transformed into z that is the result of a collision-free one-way function applied to the decryption of e: $z = \Omega(\mathcal{D}_k(e), v')$.

(3) Opening bit commitments shows that $k' = k$ and $e' = e$.

(4) Proof of knowledge shows that z is the result of a collision-free one-way function applied to x: $z = \Omega(x, v)$

And thus:

(4),(2): Because it is not possible to have the same result z when the collision-free one-way function is applied to two different values, $x = \mathcal{D}_k(e)$.

(4),(2),(3): $x = \mathcal{D}_k(e) = \mathcal{D}_{k'}(e')$.

(4),(2),(3),(1): Someone knowing x is close.

3.2 Security Requirements

The first security requirement for distance-bounding proof of knowledge protocols is a correct computation of an upper-bound of the distance between the prover and the verifier. This is expressed in the following.

Requirement 1. *If the distance-bounding proof of knowledge protocol is performed correctly, then the distance fraud has a negligible probability of success.*

In Requirement 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. Concerning the **DBPK** protocol, the following proposition holds.

Proposition 1. *The **DBPK** protocol is conformant to requirement 1.*

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 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:

$$\mathbf{P}_1 = \prod_{i=1}^{m} (P[b_i = k[i] | \delta(a_i) = 0] + P[b_i = e[i] | \delta(a_i) = 1]) = 2^{-m}$$

□

The second security requirement for distance-bounding proof of knowledge protocols consists in preventing terrorist frauds. This is expressed in the following.

Requirement 2. *If the distance-bounding proof of knowledge protocol is performed correctly, then the terrorist fraud has a negligible probability of success.*

Before analysing the **DBPK** protocol with respect to Requirement 2, we introduce the following properties.

Property 1. *Let $\Gamma : \{0,1\}^* \rightarrow \{0,1\}^*$ 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 2. *Let* $\Omega : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ *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 3. *Let* $\mathcal{E} : \{0,1\}^m \times \{0,1\}^m \rightarrow \{0,1\}^m$ *be the function such that* $e = \mathcal{E}_k(x)$*. then the following holds:*

- *Knowing* e *and* $\mathcal{E}_k$*, it is hard to find* x *without knowing* k.
- *Given* e, k*,* $x = \mathcal{D}_k(e)$ *is efficiently computable.*
- *Is is efficient to compute* $z = \Omega(x, v)$ *from the commitments on the bits of* $k[i]$ *and* $e[i]$.

Proposition 2. *If Property 1, Property 2, and Property 3 are respected, then the* ***DBPK*** *protocol is conformant to requirement 2.*

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 $(k', e') \neq (k, e)$ is: $\mathbf{P}_2 = \prod_{i=1}^{m}(P[k[i] = k'[i]|\delta(a_i) = 0] + P[e[i] = c'[i]|\delta(a_i) = 1]) = 2^{-m}$. This shows that without knowing (k, e) such that $e = \mathcal{E}_k(x)$, the probability of success of a terrorist fraud is negligible. □

The third requirement for distance-bounding proof of knowledge protocols consists in preventing any significant information about the prover's private key to be deduced from the protocol's transcripts. This requirement is expressed in the following.

Requirement 3. *If the* ***DBPK*** *protocol is performed correctly, then no significant information about* x *can be found.*

Property 4. *Showing either* $k[i]$ *or* $e[i]$ *for all* $i \in \{0, \ldots, m-1\}$ *ensures perfect secrecy of* x.

Proposition 3. *If Property 4 is respected, then the* ***DBPK*** *protocol is conformant to requirement 3.*

The fact that revealing bits of the ciphertext and bits of the key does not reveal any information on x is mandatory to enable that the whole protocol be zero-knowledge.

4 Our Scheme

This section presents a distance-bounding proof of knowledge protocol that consists of exactly the same building blocks of the **DBPK** protocol. The described protocol will be denoted **DBPK1**.

The two first phases of the **DBPK1** protocol are global settings. 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 $\mathcal{Z}_p^*$
- TA chooses an element $h \in_R \mathcal{Z}_p^*$

The randomly chosen element h will be used by the commitment algorithm. The only requirement is that neither of the prover and the verifier knows $\log_g(h)$. This either can be achieved 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.

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 $x \in_R \mathcal{Z}_{p-1} \setminus \{q\}$, 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

Assuming that the global settings have been performed, the actual distance-bounding proof of knowledge protocol can be performed by the prover P and the verifier V. In the *Bit Commitments* stage, P chooses a randomization factor u and splits the randomized private key into two secret parts k and e. Then, P performs a secure commitment on each bit of k and e.

Bit Commitments:
The following steps are performed

- P chooses $u \in_R \{0, 1, \ldots, p-2\}$ and , then sends u to V
- P chooses $k \in_R \mathcal{Z}_{p-1}$ and let $e = ux - k$
- For all $i \in \{0, \ldots, m-1\}$ where $m = \lceil log_2(p) \rceil$, P chooses $v_{k,i}$, $v_{e,i} \in_R \mathcal{Z}_{p-1}$, computes $c_{k,i} = g^{k[i]} \cdot h^{v_{k,i}}$ and $c_{e,i} = g^{e[i]} \cdot h^{v_{e,i}}$, then sends $c_{k,i}$ and $c_{e,i}$ 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 fast single challenge bit and rapid response bit exchange 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:
The following iterations are performed m times and P reveals half bits of k and e. For all $i \in \{0, \ldots, m-1\}$,

- V sends a challenge bit $a_i \in_R \{0, 1\}$ to P
- P immediately sends the response bit $b_i = \bar{a}_i k[i] + a_i e[i]$ 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 is invited, during the *Commitment Opening* stage, to open the commitments on the bits of k and e 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

- For all $i = 0, \ldots, m-1$, P sends $\bar{a}_i v_{k,i} + a_i v_{e,i}$ to V
- For all $i \in \{0, \ldots, m-1\}$, V performs the following verification: $\bar{a}_i c_{k,i} + a_i c_{e,i} \stackrel{?}{=} g^{\bar{a}_i k[i] + a_i e[i]} \cdot h^{\bar{a}_i v_{k,i} + a_i v_{e,i}}$

The *Proof of Knowledge* allows the verifier V to be sure that the sum of the secrets k and e is equal to the valuable private key of the prover P. From the bit commitments, V can compute:

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

Indeed,

$$
\begin{aligned}
k &= k[0] + 2k[1] + 4k[2] + \cdots = \sum_{i=0}^{m-1} \left(2^i \cdot k[i]\right) \mod p-1 \\
e &= \sum_{i=0}^{m-1} \left(2^i \cdot e[i]\right) \mod p-1 \text{ , and } u \cdot x = k + e \mod p-1
\end{aligned}
$$

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 $z = g^{u \cdot x} \cdot h^v$, the following proof of knowledge is performed by P and V:
$PK[(\alpha, \beta) : z = g^{u\alpha} h^\beta \wedge y = g^\alpha]$. A possible execution of such proof of knowledge is described in the following:

1. P $\longrightarrow$ V : $w_1 = g^{ur_1} \cdot h^{r_2}$ and $w_2 = g^{r_1}$ where $r_1, r_2 \in_R \mathcal{Z}_{p-1}$
2. V $\longrightarrow$ P : $c \in_R \{0, 1\}$
3. P $\longrightarrow$ V : $s_1 = r_1 - cx$ and $s_2 = r_2 - cv$
4. Verification :
   if $c = 0$: $w_1 \stackrel{?}{=} g^{us_1} \cdot h^{s_2}$ and $w_2 \stackrel{?}{=} g^{s_1}$
   if $c = 1$: $w_1 \stackrel{?}{=} z \cdot g^{us_1} \cdot h^{s_2}$ and $w_2 \stackrel{?}{=} y \cdot g^{s_1}$

As long as the verifications succeed, the interactions above are repeated t times.

5 Security Analysis

In this section we discuss the relevant security properties of the **DBPK1** protocol. First, requirements 1 and 2 are shortly analyzed and their proofs rely mainly on those of the **DBPK** (see Section 3.2). Next, properties of the encryption scheme that is used to hide the prover's private key are studied.

5.1 Preventing Distance-Bounding and Terrorist Frauds

Proposition 4. *The **DBPK1** protocol is conformant to requirement 1.*

Proof. The **DBPK1** protocol is a implementation of the **DBPK** protocol where the function δ corresponds to the identity, i.e. $\forall i\ \delta(a_i) = a_i$. This makes the proposition straightforward. (see Proposition 1) □

The second security property of the **DBPK1** protocol is expressed in the following.

Proposition 5. *The **DBPK1** protocol is conformant to requirement 2.*

Proof. The **DBPK1** consisting of the same building blocks than those of the **DBPK** protocol, then it is sufficient to prove that Property 1 and Property 2 are respected in **DBPK1**. (see Proposition 2)

(1) The function $\Gamma : x \mapsto g^x$ respects Property 1 thanks to the intractability of the *discrete logarithm* problem.

(2) The function $\Omega : (x, v) \mapsto g^x \cdot h^v$ respects Property 2 thanks to the intractability of the *representation* problem.

□

The function $z = \Omega(x, v)$ is implemented as $z = g^x \cdot h^v \mod p$ that is a representation of a value z with respect to a generator tuple (g, h). It can be shown that it is not feasible to generate two representations of the same value (see Appendix A).

5.2 Encryption of the Private Key

We now evaluate the security of the encryption scheme $\mathcal{E}$ that is used to encrypt the secret x, i.e. $e = \mathcal{E}_k(x)$. First we study whether one-time pad can be used and next we describe extensions of this basic scheme to fit the requirements.

Property 4 states that during the i^{th} distance bounding phase, the prover has to reveal the i^{th} bit of the encryption of the secret x: $e[i]$ where $e = \mathcal{E}_k(x)$; or the i^{th} bit of the key: $k[i]$. Of course revealing either $e[i]$ or $k[i]$ should not reveal anything about x.

Property 3 implies that the proof of knowledge links the knowledge of k and e to the knowledge of x. Because k and e cannot be revealed, this proof has to be based on the commitments on each bit of k and e, i.e. $c_{(k,i)}, c_{(e,i)}$.

Proposition 6. *One-time pad encryption respects Property 4*

Proof. One-time pad assure perfect secrecy: $e = \mathcal{E}_k(x) = x \oplus k$ where k is a m bit random string that is randomly chosen for each encryption.

$$P_{X|E}(X = x \mid E = e) = P_X(X = x) = 2^{-m} \quad \text{for all} \quad x, e$$

Revealing either bit i of k or bit i of e is done by choosing a random m-bits string: $s \in_R \{0, \ldots, 2^m - 1\}$

For all $i \in \{0, \ldots, m-1\}$, $r[i] = \begin{cases} \text{k[i] if s[i]=0} \\ \text{e[i] if s[i]=1} \end{cases}$ $r'[i] = \begin{cases} \text{e[i] if s[i]=0} \\ \text{k[i] if s[i]=1} \end{cases}$

Bits r are revealed and bits r' are kept secret. Like this, $r = x \oplus r'$ is a new one-time pad and thus perfect secrecy of x is still ensured. □

Unfortunately, exclusive-or does not suit modulo operations that seems mandatory when dealing with string commitments and thus does not respect Property 3. Indeed, $x = e \oplus k \mod p \neq (e \mod p) \oplus (k \mod p)$. For instance, $10 \oplus 13 = 7 \mod 11$ but $(10 \mod 11) \oplus (13 \mod 11) = 8$. Thus, another type of encryption is necessary.

Proposition 7. *One-time pad like encryption based on group addition $p-1$ respects Property 3.*

Proposition 8. *One-time pad like encryption based on addition modulo 2^m respects Property 4.*

Proof. Addition modulo can be used to implement perfect secrecy. For instance, let x be a secret in $\mathcal{Z}_n$ that is encrypted as $e = x - k \mod n$ where $k \in_R \mathcal{Z}_n$. c ensures the perfect secrecy of x:

$$P_{X|E}(X = x \mid E = e) = P_X(X = x) = \frac{1}{n} \quad \text{for all} \quad x, e$$

When r, r' are computed as previously and when $n = 2^m$, perfect secrecy of x is ensured when revealing r. In this specific case, $e, k, r, r' \in \mathcal{Z}_n$. □

However, when n is not a power of two, perfect secrecy of x is no more ensured when revealing r. In this case, $e, k \in \mathcal{Z}_n$ and $r, r' \in \{0, \ldots, 2^k - 1\}$. For instance, choosing randomly bits of $k = 0111_b \in Z_{10}$ and $c = 1011_b \in Z_{12}$ can possibly result in $r = 1111_b \notin Z_{12}$. Figure 3 shows an example of the distribution of r for different choices of x.

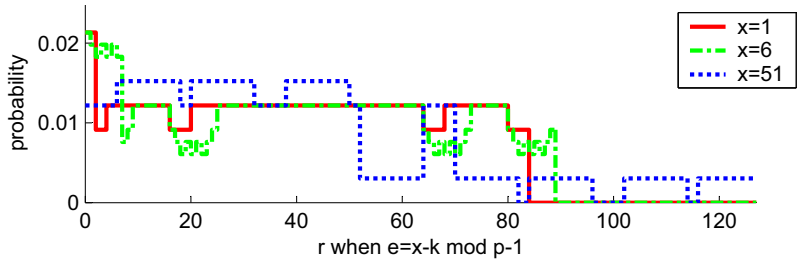


Figure 3: Distribution of r when $e = x - k \mod p - 1$, $(p = 73,\ 2^m = 128,\ x = \{1, 6, 51\})$.

Even when the key is randomly chosen before each encryption, a statistical attack aiming at retrieving x is possible (no perfect secrecy) but it is expensive. However, it is possible to get information on x (e.g. most significant bit of x). A straightforward approach to make the statistical attack more difficult is to increase the noise, i.e. define k and e in a larger set of values. Unfortunately there is an important cost in terms of number of rapid bit exchanges during the distance bounding.

Proposition 9. *One-time pad like encryption based on addition modulo a Fermat prime p respects Property 3 and Property 4.*

Proof. A *Fermat prime* [15] $p = 2^m + 1$ combines Propositions 7 and 8 and thus could solve our problem. Unfortunately there are less Fermat primes than Mersenne primes (i.e. $p = 2^m - 1$). Fermat primes can always be described as $F_n = 2^{2^n} + 1$ and only five Fermat primes are known: $\{F_0, F_1, F_2, F_3, F_4\}$. It seems unlikely that any more will be discovered soon. Anyway, F_0 to F_4 are too small to be used in the context of this paper and the next F_k prime, does it exist, would be too large (i.e $F_k >> 2^{1024}$). □

Encryption without Perfect Secrecy

Having not found an encryption scheme respecting Property 3 and Property 4, we replace Property 4 by:

Property 5. *Showing either $k[i]$ or $e[i]$ for all $i \in \{0, \ldots, m-1\}$ discloses minimal information on x.*

Here we propose an approach that makes the statistical attack more difficult without increasing the size of e and k. However, it still does not ensure perfect secrecy.

The ciphertext $e = \mathcal{E}_k(x) = u \cdot x - k \mod p$ where $u \in_R \{0, \ldots, p-2\}$ and the secret key $k \in_R \mathcal{Z}_{p-1}$ are randomly chosen before each encryption of the secret x. The parameter u is public but makes statistical analysis more difficult.

In order to ensure that r reveals minimal information on x, it is necessary that the order of subgroups created by different x be the same. In other words, it is necessary to use a strong prime: $p = 2q + 1$ where p and q are primes. It means that $\phi(p) = p - 1 = 2q$ and $\phi(\phi(p)) = q - 1$.

Efficient algorithms for finding strong primes exist. The chance of a random integer p being prime is about $1/ln(p)$ and thus the probability that p be a strong prime is about $1/(ln(p) \cdot ln((p-1)/2)) \cong 1/ln(p)^2$ when p is a m bits value, the probability that p be a strong prime is about $1/(ln(2)^2 \cdot m^2)$. In other words, finding a 1024 bits strong prime requires half a million primality verifications.

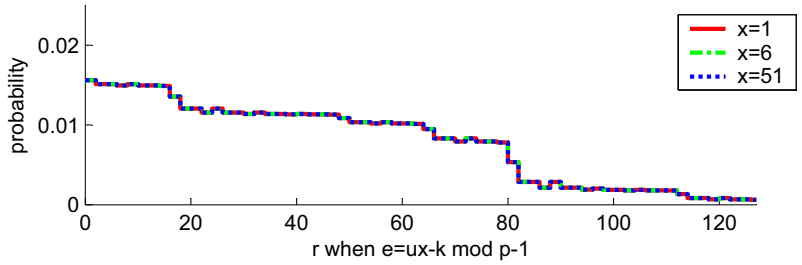


Figure 4: Distribution of r when $e = u \cdot x - k \mod p$ and p is a strong prime ($p = 83$, $2^m = 128$, $x = \{1, 6, 51\}$).

Using strong primes, the distribution of r does not reveal any information on x (see Figure 4). In fact, the distribution only depends on $p - 1$ and 2^m when x is odd and different from q. In practice, it is only necessary to avoid subgroups with very small cardinality (i.e. $x \neq q$).

Proposition 10. *One-time pad like encryption based on addition modulo $p - 1$ and multiplicative factor u respects Property 3 and Property 5 when*

$e = \mathcal{E}_k(x) = u \cdot x - k \mod p-1$ *and* p *is a strong prime, i.e.* $p = 2q+1$ *where* q *is prime.*

Proof. The number of sample necessary to retrieve information on x from r when $e = x - k \mod p-1$ is difficult to evaluate because it depends on the effect of modulo operation and random selection of bits (see Figure 3). However, we can show that at least two different r are necessary to get some information on x.

And thus, when x is encrypted as $e = u \cdot x - k \mod p-1$, it is necessary to collect at least two r for the same u. The birthday paradox shows that $2^{m/2}$ samples are necessary and thus the probability of disclosing information on x is smaller than the probability of using twice the same key k:

$$P_{\text{info on } x} < 2^{-m/2} \quad \text{where } m \text{ is the size of } r, x, k$$

□

Defeating Intruders with Partial Knowledge

The security of the scheme relies on the fact that a prover that is able to participate to the distance-bounding bits exchange has to know e and k and thus can retrieve x. However a malicious prover P could provide all bits of e and k but j random bits. For instance, $\tilde{e} = \{e_0, e_1, \ldots, \bar{e}_i, \ldots, e_{m-1}\}$ and $\tilde{k} = \{k_0, k_1, \ldots, \bar{k}_{i'}, \ldots, k_{m-1}\}$ where two bits e_i and $k_{i'}$ have been changed. Because the verifier selects randomly half bits during the distance bounding, the probability of undetected use of $\tilde{e}, \tilde{k}$ is:

$$P_{\text{undetected } \tilde{e},\tilde{k}} = 2^{-j}$$

and the probability of I finding x is:

$$P_{I \text{ gets } x} = (2 \cdot m)^{-j}$$

For instance, with $m = 1024$ bits and $j = 6$, 32 tries are necessary to let I impersonate P and the probability that I discovers x is 2^{-66}. This is a marginal attack but we propose that V returns an error message when a bit is false in order to help a potential intruder to find out x. Thus, P avoids to provide e and k or even $\tilde{e}$ and $\tilde{k}$ to another party.

Conclusion and Further Work

In this paper, we addressed the problem of terrorist frauds in application scenarios where cryptographic identification 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. We first presented a general scheme that shows the main components of such protocols. We then presented a possible implementation of such protocols and analyzed its security properties. Even though we have not reached perfect zero-knowledge, 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 $\mathcal{E}$, and representation Ω exist. We proposed a solution relying on a public key scheme based on the discrete log problem, bit commitment based on discrete log, group addition one-time pad, and representation problem: $DBPK1 = DBPK[\alpha : y = g^{\alpha}]$. This scheme could directly be used with ElGamal's and Schnorr's identification schemes that both rely on the discrete log problem.

The integration of distance bounding with Fiat-Shamir identification scheme is not straightforward. The public key x is chosen in $\mathcal{Z}_n$ where $n = pq$ and the public key is $x^2 \mod n$. It is necessary to define $DBPK[\alpha : y = \alpha^2]$. And, 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, 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. For instance, it could be integrated in a group signature scheme. For instance [8] could be used as follows:

$$\begin{array}{l} \text{DBPK}[\alpha \mid \tilde{z} = \tilde{g}^{(a^{\alpha})}] \\ \text{PK}[\beta \mid \tilde{z}\tilde{g} = \tilde{g}^{(\beta^e)}] \end{array}$$

The verifier can thus 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 next step will be the integration of distance bounding protocols in unlinkable and/or pseudonymous credential schemes such as Idemix [7].

Even if the proof of knowledge is zero-knowledge, the whole scheme cannot be done zero knowledge because the same k could appear twice (with probability $2^{-m/2}$) and thus V could get all bits of k and e and compute x. In

other words, even with an encryption scheme ensuring perfect secrecy of x and a zero knowledge proof of knowledge, only statistical zero knowledge can be achieved. We are still working on other approaches based on probabilistic encryption of x to ensure that the whole protocol be zero knowledge.

Terrorist fraud attacks can be forbidden by using Chaum's distance bounding protocol combined with a tamper-resistant hardware that is certified [6]. In this case, the verifier can check that k and e were generated by a trustworthy (i.e. certified by a TTP) hardware that will not disclose those secrets to an intruder. Thus it is equivalent to our DBPK protocol. However, our scheme is more general and is easier to deploy because it does not rely on tamper-resistant hardware or device certification.

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] Stefan Brands. An efficient off-line electronic cash system based on the representation problem. Technical report, 1993.

[6] 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.

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

[8] 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.

[9] 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.

[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 and K. Zhang. Validating and securing spontaneous associations between wireless devices. In *Proceedings 6th Information Security Conference (ISC03)*, volume 765, pages 44–53, 2003.

[14] 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.

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

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

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

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

A The Representation Problem

This section proves that it is not possible to generate two representation of a value z with respect to a generator tuple (g, h), i.e. two pairs (a_1, a_2) and (a'_1, a'_2) so that $g^{a_1}h^{a_2} = g^{a'_1}h^{a'_2}$. Generator g and h are chosen randomly so that the logarithm of h to the base g cannot be computed. In our scheme, it means that g and h cannot be chosen by the prover. The remaining of this section is a simplification of the proof that can be found in [5].

Proposition 11. *The following statements are equivalent.*

(1) There exists a polynomial-time algorithm $A_{(1)}$ which, on input a generator tuple (g, h), output a number z and two different representing index tuple of z: $A_{(1)}(g, h) \to z, (a_1, a_2), (a'_1, a'_2)$ with $z = g^{a_1}h^{a_2} = g^{a'_1}h^{a'_2}$.

(2) There exists a polynomial-time algorithm $A_{(2)}$ which, on input a generator tuple (g, h), output a nontrivial representing index tuple of 1: $A_{(2)}(g, h) \to (a_1, a_2)$ with $1 = g^{a_1}h^{a_2}$.

(3) There exists a polynomial-time algorithm $A_{(3)}$ which solves the Discrete Log problem.

Proof. We only need to show probabilistic polynomial-time transformations of 3) to each of 1) and 2), since we can come up easily with feasible algorithms $A_{(1)}$ and $A_{(2)}$ if we have $A_{(3)}$.

(1) ⇒ (2) Algorithm $A_{(2)}$ proceeds as follows:

1. Feed the generator tuple (g, h) into $A_{(1)}$ and receive z and two representing index tuples (a_1, a_2) and (a'_1, a'_2) of z.
2. Output $(a_1 - a'_1, a_2 - a'_2)$. Like this, $g^{a_1-a'_1} \cdot h^{a_2-a'_2} = z/z = 1$.

(2) ⇒ (3) Algorithm $A_{(3)}$ proceeds as follows:

1. Generate a 2-tuple (u_1, u_2) at random, and compute the generator-tuple (g, h) according to $g = a^{u_1}, h = b^{u_2}$.
2. Receive an index-tuple (a_1, a_2) from $A_{(2)}$.
3. Compute and output $\log_b(a)$: $g^{a_1} \cdot h^{a_2} = 1 = b^0$ or $a^{u_1a_1} \cdot b^{u_2a_2} = b^0$ and thus $\log_b(a)u_1a_1 + u_2a_2 = 0$

$$\log_b(a) = -\frac{u_2a_2}{u_1a_1}$$

□