Simple PIN recovery attack on a standardised financial transaction protocol

Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
`C.Mitchell@rhbnc.ac.uk`

24th November 1999

Abstract

A standardised method of encrypting PINs, for use when a PIN is sent to or from an ICC for verification, is shown to be flawed. This flaw will typically enable the PIN to be discovered from its encrypted version in a small number of trials.

Keywords: PIN, encryption, banking, security.

1 Introduction

A recently published international standard, [2], describes a range of protocols for secure transactions using Integrated Circuit Cards (ICCs, often known as 'smart cards'). Typically such protocols would be used between an ICC and a terminal.

One of these protocols, described below, is designed to be used to enable the verification of an ICC Personal Identification Number (PIN), i.e. a secret password held by the ICC user which is typically a four-digit number. One likely application of the protocol would be where a user enters their ICC into a terminal, and at the same time types their PIN into a keypad attached to the terminal. The protocol allows the terminal to pass the PIN in encrypted form to the ICC for verification.

2 The ‘PV-asymmetric’ PIN verification protocol

The ‘PV-asymmetric’ PIN verification protocol is specified in Clause 6.6.3 of [2]. In this protocol, which consists of two messages, a public key certificate is passed from entity B to entity A, and A returns an encrypted PIN block.

The standard does not specify exactly how this protocol is to be used, although there are two obvious possibilities. In one application, which we might call ‘on-card PIN verification’, entity A might be a terminal and entity B an ICC. In this case the protocol could be used to encrypt a PIN, entered by a user into the terminal, for secure transfer to the ICC where it can be verified. In another application, which we might call ‘off-card PIN verification’, entity A might be an ICC and entity B a terminal. In this case the card might pass the encrypted PIN to the terminal for comparison with a PIN offered by the user.

In more detail the security-critical components of the two messages are as follows.

$$\begin{aligned} B \rightarrow A: &\quad \mathrm{CID}_B || \mathrm{Cert}_B \\ A \rightarrow B: &\quad \mathrm{KID}_{P_B} || eP_B(\mathrm{PBF1}) \end{aligned}$$

The following notation is used:

- CID_B is the certificate identifier for Cert_B,
- Cert_B is a certificate for B's public key P_B, signed by a Certification Authority whose public verification key is possessed by A,
- KID_{P_B} is a key identifier for the public key P_B,
- $eP_B(X)$ denotes the public key encryption of data X using the public key P_B, and
- PBF1 is a PIN block, containing a secret PIN, constructed using PIN block format 1 (as defined in ISO 9561–1, [1]).

A PIN block produced according to PIN block format 1 (see clause 8.3.2 of [1]) is a 64-bit block. It is produced as an encoding of the following values:

- a PIN of between 4 and 12 decimal digits,
- the *PIN length*, a value N in the range 4–12 inclusive, and
- a *transaction field* of $56 - 4N$ bits (i.e., if the PIN has 4 digits, then this field will have 40 bits).

It is suggested in clause 6.6.3 of [2] that the transaction field should be set equal to a non-recurring counter value or the date/time. This is to protect against an 'enumeration attack'. This is not defined, although it is presumably a reference to the fact that, if the transaction field is a constant value, then it would be straightforward to derive a 'dictionary' of PIN values against their corresponding encrypted values.

The processing associated with the protocol is straightforward. When A receives the first message, A verifies the certificate, and, using this certificate, decides whether B is an entity to which it is prepared to send the PIN. A then assembles the PIN block, encrypts it using the public key obtained from the certificate, and then sends it to B.

3 PIN recovery from an encrypted PIN block

Unfortunately the method of using encryption described in the above protocol is insufficient to protect the secrecy of the PIN. This is because, using the terminology of page 288 of [3], a 'forward search' attack is simple to conduct, regardless of the encryption algorithm. As described in [3], a forward search attack is possible when the plaintext space is small or predictable, in which case the adversary can decrypt a ciphertext c by simply encrypting all possible plaintexts until c is obtained.

In the above protocol, any party observing the pair of messages can conduct such an attack. The public key sent in the first message can be used to encrypt candidate values for the PIN block and compare them with the actual value sent in the second message. If the advice in the standard is followed, then the only variable values in the PIN block will be the PIN (typically of 4 decimal digits) and either a counter value or a date/time stamp. If a counter value is used, and the counter starts at zero, then when the system is started the attacker will know the transaction field value. Alternatively, if a date/time stamp is used, the attacker will know the value of the transaction field lies among a small number of possibilities.

To assess the power of the attack we suppose that the attacker can encrypt 10 candidate blocks per second (not an unreasonable assumption for a PC implementation), that PINs have four digits, and that the attacker knows that the transaction field is one of 100 possibilities. Then the attacker will immediately know that there are only 10^6 possibilities for the PIN block, and working through all the possibilities will take approximately one day.

In fact, even if the system is implemented using a completely random transaction field, then, if PINs have four digits, there will still only be approximately $10^4 \times 2^{40} \simeq 10^{16}$

possibilities for the PIN block. If an attacker has access to a large number (say 10^4) of such encrypted PIN blocks, all encrypted using the same public key, then a search of size 10^{12} has a good chance of yield at least one plaintext PIN.

Given the sensitivity of PINs, this type of situation is very unlikely to be acceptable in practice, and the standard is clearly in need of revision. One might argue that intercepting encrypted data exchanged between an ICC and a terminal is difficult. However it is not difficult to design a device which intercepts data sent between a smart card and a terminal; moreover, if data interception between ICC and terminal is not considered a potential threat, then it is curious that a method for encryption is specified in the standard.

Fortunately there is a very simple, and well-known, remedy to the problem. It is to pad the message to be encrypted with a long string of random bits prior to encryption. There should be plenty of scope for this, since the block length of public key encryption schemes is typically much longer than 64 bits. Such a recommendation should be added to the standard. Finally note that exactly similar problems to those described above arise with the protocol specified in clause 6.6.4 of [2], and similar changes are required.

4 Other shortcomings of the standard

Before concluding note that there are a number of other security shortcomings in [2].

- Clauses 6.4.3 and 6.4.4 describe general purpose protocols for transferring secret messages using public key encryption. No guidance is given on the need for random padding when the messages are predictable.

- Some of the key exchange mechanisms specified in clause 6.1 are subject to possible 'replay' attacks.

- Some of the entity and message authentication mechanisms specified in clauses 6.2 and 6.3 are subject to 'preplay' and 'reflection' attacks.

5 Conclusions

A major weakness has been described in a protocol in a recently published international standard. If the defective mechanism is deployed, then the secrecy of user PINs could easily be compromised.

References

[1] International Organization for Standardization, Genève, Switzerland. *ISO 9564-1: 1991, Banking — Personal Identification Number management and security — Part 1: PIN protection principles and techniques*, December 1991.

[2] International Organization for Standardization, Genève, Switzerland. *ISO 10202-5: 1998, Financial transaction cards — Security architecture of financial transaction systems using integrated circuit cards — Part 5: Use of algorithms*, July 1998.

[3] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, 1997.