Cryptanalysis of a recent two factor authentication scheme

Michael Scott

Certivox Labs
Invent Centre
Dublin City University
Ballymun, Dublin 9, Ireland.
mike.scott@certivox.com

Abstract. Very recently a scheme has been proposed by Wang and Ma for a robust smart-card based password authentication scheme, which claims to be secure against a Smart Card security breach. In this short note we attempt an initial cryptanalysis of this scheme.

1 Introduction

Two factor authentication schemes which support a range of desirable attributes, have proven difficult to develop. In [1] two existing schemes are successfully cryptanalysed, and a new scheme is proposed. Here we analyse this new scheme and point out some problems with it.

In one sense it is not hard to develop a two factor authentication scheme based on a password and on possession of a token. One might for example combine a traditional Username/Password scheme with a completely distinct process based on a large cryptographically secure shared secret stored on a token and also on the server. However such a scheme would not be regarded as satisfactory, as if the server were hacked the system would be completely broken with potentially very damaging consequences for the service provider.

Therefore typically designers of such schemes work against a self-imposed list of desirable attributes and the scheme of [1] is no exception. They list twelve desirable attributes numbered from C1 to C12. They also list the five capabilities of an adversary. At first glance the attributes are very compelling, and the adversary appears to be granted impressive powers. The authors argue convincingly that their twelve attributes are met, even in the face of their most capable adversary.

However here we point out that while technically the authors may be correct, in a real-world context an adversary can break the scheme by assuming some rather less impressive capabilities that are strictly speaking not allowed to him. Also the list of desirable attributes is shown to be deficient.

2 A Deconstruction of the Wang-Ma scheme

The full description of the scheme can be found in the original paper, and we will not repeat it here. We also omit some details that are not relevant to our analysis. Basically the scheme is based on a smart-card, access to which is protected by a password. The smart card is issued to the client by the server, and the client also enters into it some masking values of their own. On the smart card is stored a password-protected cryptographically strong long-term secret of the form $H(x\|ID_j\|T_{reg})$, where $H(.)$ is a hash function and x is the server's secret key. Note that the long-term secret can be derived by the server if they know the static identity of the client (ID_j) and the time of their first registration (T_{reg}). To this end a table $\{ID_j, T_{reg}\}$ of registered identities and the time of their registration is maintained by the server. The client's long-term secret is effectively doubly-protected by the password so that breaching the smart card security still does not reveal it. The clever means by which this is achieved is not relevant here.

The first attribute C1 of the proposed scheme demands that passwords or derived values should not be stored on the server. The reason is that servers are commonly hacked, and the loss of such a file is catastrophic. In [1] this is achieved as the password is only relevant for communication between the client and their smart-card – it is not needed at all by the server. This simple idea immediately obtains the first three listed desirable attributes (nothing password related stored on the server, the server cannot derive the password, and the password can be changed locally).

Note however that while nothing password related is stored on the server, identity-based information is stored by the server directly in the table $\{ID_j, T_{reg}\}$. This table has no cryptographic protection.

One of the more interesting attributes is that of user anonymity (C11). In this context this means that the actual client identity is never transmitted in the clear during a protocol run. We note is passing that any such scheme can achieve this property by the simple expedient of running under the tried-and-tested SSL protocol. But a careful reading of [1] seems to suggest that the secrecy of the identity rather than being a deliverable of the scheme, is in fact a requirement for its security. We find this to be a highly dubious strategy. Identities, which have a structure, are not commonly assumed to have cryptographic strength. They are usually entered in the clear and hence vulnerable to "shoulder-surfing". They are easily guessed. And clients while used to the idea of keeping passwords a secret would not normally be expecting to have to keep their identity a secret as well. Nevertheless it appears that in [1] identities are used as a kind of surrogate extra password, shared between the client and the server.

So while no passwords or derived details are stored on the server, identities are. Of course there is nothing in the list of desirable properties to discourage this.

We observe that the ability to hack into the server and access this file of secret identities is not allowed to the adversary in [1]. Overall we regard the restriction that the adversary does not know the identity of the client as being

unreasonable. After all an adversary who has the ability to access a clients smart card and extract its secrets should surely be able to determine their identity, which by the authors own admission is of a "predefined format" and "easy to remember".

We will proceed on the basis that an adversary would be able to determine the clients identity, though we admit that this is not strictly allowed for in [1].

3 Cryptanalysis

Cryptanalysis now proceeds quite easily. An adversary captures the smart card belonging to ID_i and extracts its secret information, as allowed for the adversary as described in [1]. The adversary then logs into the Server, ignoring steps L1 and L2, generating random u, calculating $Y_1 = y^u \bmod p$ where y is the server's public key, and sending to the Server the pair $C_1 = g^u \bmod p$ and $CID_i = ID_i \oplus H(C_1 \| Y_1)$.

The server responds correctly by calculating $Y_2 = Y_1 = {C_1}^x$ and recovers $ID_i = CID_i \oplus H(C_1 \| Y_2)$. Since ID_i is valid the server goes on to generate random v and computes $K_S = {C_1}^v$, $C_2 = g^v \bmod p$, and $k = H(x \| ID_i \| T_{reg})$ and $C_3 = H(ID_i \| ID_S \| Y_2 \| C_2 \| k \| K_S)$. The server then sends C_2 and C_3 back to the adversary, who then drops the connection.

The adversary first computes $K_U = K_S = {C_2}^u \bmod p$. Finally, at his leisure, he guesses a value for the password and recovers a candidate value k' for the long-term secret. He then calculates ${C_3}^* = H(ID_i \| ID_S \| Y_1 \| C_2 \| k' \| K_U)$. If C_3 equals ${C_3}^*$ then the password guess was correct, otherwise try again. This is a classic off-line dictionary attack, the downfall of many such schemes. In this way once the first factor (the smart-card) is compromised, so is the second (the password), and the scheme is no longer two-factor.

4 Discussion

We note that the proposed scheme has a proof of security. But a proof is only as good as its assumptions, and an adversary is not necessarily bound by any assumptions. An adversary is also not bound by any list of supposed capabilities.

We would suggest that the use of identities as a kind of surrogate password is not a viable strategy for the development of such schemes. It must be assumed that a determined adversary should have no trouble in determining the static identity of a client. If password derived data should not be stored on the server, and if identities are to be used as a kind of extra password, then identity-derived information should also not be stored by the server, and this should form one of the desirable attributes of such a scheme.

References

1. D. Wang and C. Ma. Robust smart card based password authentication scheme against smart card security breach. Cryptology ePrint Archive, Report 2012/439, 2012.