Cryptanalysis of the Wu-Varadhrajan Fair Exchange Protocol

Olivier Markowitch and Shahrokh Saeednia

Université Libre de Bruxelles,
Département d'Informatique,
Boulevard du Triomphe - CP212
1050 Bruxelles, Belgium

{omarkow,saeednia}@ulb.ac.be

Abstract. In this paper we present an attack on a fair exchange protocol proposed by Wu and Varadharajan. We show that, after two executions of the protocol, a dishonest participant can collect enough information in order to obtain some secret information of the other participant. This precisely allows him to compute the final signature of the other participant in all subsequent executions of the protocol, without disclosing his own signature.

Keywords: cryptography, digital signatures, fair exchange protocols

1 Introduction

A fair exchange protocol consists of two entities that wish to exchange digital signatures in such a way that at the end of the protocol either both signatures are correctly exchanged or none of them is indeed received.

Briefly, a fair exchange protocol works in the following way: in the first phase of the protocol, the entities exchange committed signatures, whose validity ensure the recipient that in case of problem a Trusted Third Party (TTP) will have enough information to convert the committed signature into another one that is called a "final signature". Once a committed signature is succesfully verified, the entities exchange their final signatures in the second phase.

In a normal situation the entities communicate directly between them, but in case of problems (when an expected message is not communicated during the protocol) each entity can request the help of the TTP, which will conclude the protocol in a fair way. In this case, the TTP is said to be offline.

Classically, fair exchange protocols with offline TTPs are such that the TTP provides, in some circumstances, its own signature as an affidavit that has the same legal value than the signature of the entities [1]. There are, however, protocols [2, 3] in which the signature that the TTP provides is instinguishable from that produced by the entities in a fault-free case. In this case, the TTP is called invisible. This means that, at the end of the protocol, by only observing the produced signatures, it is impossible to decide whether the TTP has intervened in the execution of the protocol. As the intervention of the TTP can be due to a network failure rather than a cheating party, invisible TTPs are very useful in the some contexts (e.g., electronic commerce), in order to avoid confusing reputations for the parties.

Recently, Wu and Varadharajan proposed [4] a fair exchange protocol of this type that is based on the DSS signature scheme. In this paper, we show that the Wu-Varadhrajan protocol is not secure. We present an attack allowing a dishonest entity involved in two executions of the protocol to obtain enough information in order to generate the final signature of the other entity in all subsequent executions of the protocol without revealing his own signature. Consequently, the protocol is not fair.

2 The Wu-Varadhrajan fair exchange protocol

The protocol is based on the DSS signature scheme. After a successful execution of the protocol, both entities obtain a classical DSS digital signature provided by the other entity.

2.1 The DSS

During the setup phase, the TTP chooses two large public primes p and q such that q is a large factor of $p-1$, and a public generator g for the subgroup of $\mathbb{Z}_p^*$ of order q.

The signer chooses randomly an integer $x \in \mathbb{Z}_q$ as his private key and computes $y = g^x \bmod p$ as the corresponding public key.

Signature generation. The signer randomly chooses $k \in \mathbb{Z}_q^*$, computes $r = \left(g^k \bmod p\right) \bmod q$ and $s = k^{-1}\left(h\left(m\right) + xr\right) \bmod q$, where h is a public hash function. The signature is the pair (r, s).

Signature verification. The verifier computes $u_1 = h\left(m\right)s^{-1} \bmod q$ and $u_2 = rs^{-1} \bmod q$. The signature is accepted as valid if $r \equiv \left(g^{u_1}y^{u_2} \bmod p\right) \bmod q$.

2.2 Committed signature

In the context of the Wu-Varadhrajan fair exchange protocol, an additional setup is needed for the committed signatures.

Each user and the TTP agree on two random secrets v and $w \in Z_q$. The user is also given the following public information issued by the TTP: $\gamma = g^v \bmod p$, $\gamma' = g^{v^{-1}} \bmod p$ and $\lambda = \gamma'^{w\gamma} \bmod p$.

Committed signature generation. For a committed signature, the signer first computes r and s as a DSS signature and forms $u_1 = h(m)\, s^{-1} \bmod q$ and $u_2 = rs^{-1} \bmod q$. Then the signer computes $v' = v^{-1}(h(m) + w\gamma) \bmod q$, $\alpha = g^{u_1} \bmod p$, $\beta = y^{u_2} \bmod p$, $\delta = y^{v'} \bmod p$, $r = (\alpha\beta \bmod p) \bmod q$, $c = r^{-1}h(m) \bmod q$, $e = h(\alpha, \beta, \delta, c)$ (where "," denotes the concatenation) and $z = (v' + eu_1) \bmod q$. The committed signature is $(\alpha, \beta, \delta, z)$.

Committed signature verification. The verifier computes $r = (\alpha\beta \bmod p) \bmod q$, $c = r^{-1}h(m) \bmod q$ and $e = h(\alpha, \beta, \delta, c)$. The committed signature is accepted if $g^z \equiv \gamma'^{h(m)}\lambda\alpha^e \pmod p$ and if $y^z \equiv \delta\beta^{ce} \pmod p$.

2.3 Final signature

After the exchange of the committed signatures and succesful verification by the entities, they exchange their final digital signature consisting of their respective DSS signatures (i.e., r and s).

Note that, the final signatures can also be provided by the TTP, thanks to its knowledge of v and w. Knowing the committed signature $(\alpha, \beta, \delta, z)$ of one entity communicated by the other entity, the TTP can compute

$$r = (\alpha\beta \bmod p) \bmod q$$

$$c = r^{-1}h(m) \bmod q$$

$$s = h(m)\, h(\alpha, \beta, \delta, c) \left(z - v^{-1}(h(m) + w\gamma)\right)^{-1} \bmod q$$

3 The attack

During an instance of the protocol, the committed signature on a message m is $(\alpha, \beta, \delta, z)$. The recipient of such a committed signature can compute:

$$r = (\alpha\beta \bmod p) \bmod q$$

$$c = r^{-1}h(m) \bmod q$$

$$e = h(\alpha, \beta, \delta, c)$$

Now, when this committed signature is converted into the corresponding final signature (by the signer or by the TTP), the recipient knows the related u_1 and is able to compute $v' = z - eu_1$

If the protocol is executed again for another message $\tilde{m}$, the recipient will be able to compute $\widetilde{v'} = \tilde{z} - \tilde{e}\tilde{u}_1$, in the same way.

So, the recipient can compute

$$a = v'\widetilde{v'}^{-1} \mod q$$

and the following equation holds: $a = (h(m) + w\gamma)(h(\tilde{m}) + w\gamma)^{-1} \mod q$.

Consequently, we have:

$$h(\tilde{m})a + wa\gamma = h(m) + w\gamma \mod q$$

$$w(a\gamma - \gamma) = h(m) - h(\tilde{m})a \mod q$$

$$w = (h(m) - h(\tilde{m})a)(a\gamma - \gamma)^{-1} \mod q$$

Since γ is public, the recipient can compute w and subsequently v as

$$v = v'^{-1}(h(m) + w\gamma) \mod q \tag{1}$$

Therefore, after two complete executions of the Wu-Varadhrajan protocol, the recipient knows v and w (modulo q) of the signer. With this information, the recipient of a committed signature is able to convert it into a final signature, exactly in the same way as the TTP does.

Note that, knowing these values modulo q is not restrictive because v and w are always used modulo q in the protocol.

It could be argued that the parameter γ is to be kept secret in order to make this attaque ineffective. However, even in this case the protocol remains insecure. Indeed, when generating committed signatures, γ is always considered multipled by w, and it suffices to have $w\gamma$ (or more properly, $w\gamma \mod q$) rather than w. So, we may compute

$$w\gamma = (h(m) - h(\tilde{m})a)(a - 1)^{-1} \mod q$$

and v as in equation 1.

4 Conclusion

We have presented an attack on the the fair exchange protocol of DSS-like signatures of Wu and Varadharajan. We showed that, after two executions of the protocol, a dishonest participant is able to obtain the secret information of the other participant, allowing him to get, for all subsequent executions of the protocol, a final signature without revealing his own digital signature.

References

1. N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *Proceedings of the fourh ACM Conference on Computer and Communications Security*, pages 8–17. ACM Press, Apr. 1997.
2. N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. In *Advances in Cryptology: Proceedings of Eurocrypt'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 591–606. Springer-Verlag, 1998.
3. O. Markowitch and S. Saeednia. Optimistic fair-exchange with transparent signature recovery. In *5th International Conference, Financial Cryptography 2001*, Lecture Notes in Computer Science. Springer-Verlag, 2001.
4. C.-K. Wu and V. Varadharajan. Fair exchange of digital signatures with offline trusted third party. In *Information and Communications Security. Third International Conference, ICICS 2001*, volume 2229 of *Lecture Notes in Computer Science*, pages 466–470. Springer-Verlag, 2001.