University of Technology Chemnitz-Zwickau

Department of Computer Science

Theoretical Computer Science and Information Security

Technical Report

TR-95-2

Comment: Digital signature with (t, n) shared verification based on the discrete logarithm problem

Patrick Horster · Markus Michels · Holger Petersen

June 1995

Straße der Nationen 62 Phone: (+49) 371-531-1479

D-09111 Chemnitz, Germany Fax: (+49) 371-531-1810

Comment

Digital signature with (t, n) shared verification based on discrete logarithms[1]

Patrick Horster · Markus Michels · Holger Petersen

Indexing term: Cryptography

The digital signature scheme with (t, n) shared verification proposed by Harn [1] can be easily forged universally [2]. In his reply Harn presented two different solutions to overcome this attack [3]. We discuss the first solution and show that the second solution (and related ones) are flawed, because with a simple attack we can forge this scheme and related ones universally. Finally we propose a simple solution to countermeasure all attacks. We assume that the reader is familar with the scheme in the notation of [1, 3] and therefore we skip a review.

The first solution suggested by Harn to overcome the attack is to choose the parameters g and β in the initialization and publish them as public keys, certified by the trusted authority. Then the attack mentioned above is avoided, but for each signature new parameters g and β are necessary. Thus the disadvantage of this approach is that the size of the public file depends on the number of signatures the signer might be sign in future. As a result this countermeasure is not practical.

The second solution suggested by Harn to overcome the attack is to use the AMV- [5] instead of the ElGamal signature scheme [4] as underlying conventional signature scheme. The signature for the message m is then given by (z, r, g, β) and the verification is done by checking the congruence

$$\beta^{H(m)} \equiv y^z r^r \pmod{p}.$$

The public key y can be computed by at least t out of n verifiers and the relation $y \equiv g^{\alpha^s} \pmod{p}$ holds, where α is a fixed secret parameter and s is the secret key of the signer. We assume that the signature (z, r, g, β) for the message m is known. To obtain a (forged) signature for the message $\tilde{m}$ the attacker computes $u := H(\tilde{m})^{-1}H(m) \pmod{w}$ and $\tilde{\beta} := \beta^u \pmod{p}$. Now the signature for the message $\tilde{m}$ is then given by $(z, r, g, \tilde{\beta})$, because

$$\tilde{\beta}^{H(\tilde{m})} \equiv \beta^{uH(\tilde{m})} \equiv \beta^{H(m)} \equiv y^z r^r \pmod{p}.$$

Note that this attack (and also the attack of [2]) might be noticed by those verifiers who know all previous signatures of the signer. As the parameter r is not modified in the forged signature, the verifierers can reject those signatures where the parameter r was used before. However this assumption is not realistic and therefore the scheme is insecure. Obviously the described attack can also be used to forge a signature in the original scheme [1]. Note that the attack doesn't work in the conventional signature schemes [4, 5], if the (fixed) generator is certified by the trusted authority. Another insider attack on both schemes for an attacker who knows a pair (g, y) is to choose r and s at random and solve the verification equation for the parameter β.

It is an interesting question if the resulting scheme is still insecure if another variant of the Meta-ElGamal signature scheme [6, 7] is used as underlying signature scheme. The answer is yes, as we will see in the following: The signature for message m is (z, r, g, β), where $r' := \beta^k \pmod{p}$, $r := d(r', H(m))$ using a suitable function d, the congruence $A \equiv sB + kC \pmod{w}$ is solved for parameter z where the coefficients A, B and C can be chosen as suitable functions e,f and l with arguments $H(m), z, r$. Furthermore $g^{\alpha^s} \equiv \beta^s \pmod{p}$ holds. The signature can be verified by checking $r = d(\beta^{AC^{-1}} y^{-BC^{-1}} \pmod{p}, H(m))$, where $y \equiv g^{\alpha^s} \pmod{p}$ is computed by the t verifiers. But however, as an attacker can influence β and y, *all* schemes can be universally forged. If the signature (z, r, g, β) for the message m is known then the equation $r' \equiv \beta^{AC^{-1}} y^{-BC^{-1}} \pmod{p}$ holds and the parameter r in A, B and C can be substituted by $d(r', H(m))$. We fix the notations $\tilde{A}, \tilde{B}$ and $\tilde{C}$ which results from A, B and C respectively, if $H(m)$ is substituted by $H(\tilde{m})$, r by $\tilde{r}$ and z by $\tilde{z}$. To obtain a signature $(\tilde{z}, \tilde{r}, \tilde{g}, \tilde{\beta})$

[1] appeared in Electronics Letters, Vol. 31, No. 14, July, (1995), p. 1137.

for the message $\tilde{m}$, choose $\tilde{z}$ at random, compute $\tilde{r} := d(r', H(\tilde{m}))$, $\tilde{\beta} := \beta^{AC^{-1}\tilde{A}^{-1}\tilde{C}} \pmod p$ and $\tilde{g} := g^{BC^{-1}\tilde{B}^{-1}\tilde{C}} \pmod p$. Thus

$$\tilde{r} = d(r', H(\tilde{m})) = d(\beta^{AC^{-1}} y^{-BC^{-1}} \pmod p, H(\tilde{m}))$$
$$= d(\tilde{\beta}^{\tilde{A}\tilde{C}^{-1}} \tilde{y}^{-\tilde{B}\tilde{C}^{-1}} \pmod p, H(\tilde{m})),$$

because $\tilde{y} \equiv \tilde{g}^{\alpha^s} \pmod p$. Therefore the digital signature scheme with (t, n) shared verification is insecure with *any* other possible signature scheme. The design problem of the scheme is that the parameters g and β are not authentic and thus could be chosen arbitrarily by an attacker.

This can be prevented if the value $H(\beta, y)$ is additionally signed by the signer using any conventional signature scheme. Thus y (and therefore g) and β are authentic. This signature can only be verified after the t verifiers have computed y. Any unauthorized verifier can't check the validity of this additional signature, as he doesn't know y. Obviously, if this countermeasure is used and the functions e, f and l are chosen such that the underlying signature scheme is secure (see [6, 7] for details) then a (t, n) shared verification signature scheme can also be built using other conventional signature schemes. If such a conventional signature scheme is chosen properly, e.g. $d(r', H(m)) = r'$, $A = e(s) = s$, $B = f(H(m), r) = H(m) \oplus r$ and $C = 1$, where $\oplus$ denotes bitwise XOR and an efficient conventional signature scheme is used to authenticate y and β, then the computational costs for signature generation and verification are still reasonable.

Patrick Horster, Markus Michels and Holger Petersen
Theoretical Computer Science and Information Security
University of Technology Chemnitz-Zwickau,
Straße der Nationen 62, D-09111 Chemnitz, Germany

References

[1] L.Harn, "Digital signature with (t, n) shared verification based on discrete logarithms", Electronics Letters, Vol.29, No. 24, (1993), pp. 2094–2095.

[2] W.-B.Lee, C.-C.Chang, "Comment: Digital signature with (t, n) shared verification based on discrete logarithms", Electronics Letters, Vol.31, No.3, (1995), pp. 176–177.

[3] L.Harn, Reply to [2], Electronics Letters, Vol.31, No. 3, (1995), pp. 177.

[4] T.ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, Vol. IT-30, No. 4, July, (1985), pp. 469–472.

[5] G.B.Agnew, R.C.Mullin, S.A.Vanstone, "Improved digital signature scheme based on discrete exponentiation", Electronics Letters, Vol. 26, (1990), pp. 1024–1025.

[6] P.Horster, M.Michels, H.Petersen, "Meta-ElGamal signature scheme", Proc. 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, Nov. 2–4, (1994), pp. 96–107.

[7] P.Horster, M.Michels, H.Petersen, "Generalized ElGamal signatures for one message block", Post-Workshop Proc. IT-Sicherheit'94, Vienna, Austria, Sep. 22–23, 1994, R.Oldenbourg Wien München, (1995), pp. 66–81.