WILDCARD ATTACKS ON DENIABLE AUTHENTICATION

JEFFREY BURDGES AND CHRISTIAN GROTHOFF

Abstract. We construct a deniable authentication scheme that does not suffer from Dominic Tarr's Wildcard attack on Triple Diffie-Hellman.

We consider an authentication scheme to be *deniable* if the intended participants themselves can be confident in the authenticity of the messages they excahnge, but cannot proved authenticity to a third party after the conversation.

As an example, the Triple Diffie-Hellman component of Trevor Perrin's Axolotl ratchet provides authentication with Diffie-Hellman key exchanges, as opposed to signature operations.[] Triple Diffie-Hellman is deniable because no signatures are ever produced.

1. Wildcard attack

... Not writing much here yet as Dominic Tarr might agree to merge this paper with his existing paper ...

Axolotl itself is only vulnerable to a wildcard attack when the ratchet is initially started, after that the ratchet state itself prevents such attacks.

2. Deniable ECDSA

We suppose that E is an eliptic curve group and let G denote our base point. Also suppose that $n = |G|$ is prime and set $l = \lceil \log_2 n \rceil$. We suppose as well that $\text{hash}(\cdot)$ is a cryptographi hash function.

2.1. **Signing.** We assume that Alice has a key pair (d_A, Q_A) with private key $d_A \in [1, n-1]$ and public key $Q_A = d_A \times G$. We suppose additionally that Alice and Bob have securely communicated a random integer x in $[0, n-1]$.

Alice signs a message m as follows.

(1) Let z denote the leftmost l bits of $\text{hash}(m)$ regarded as a number.
(2) Generate a cryptographically secure random integer k in $[1, n-1]$.
(3) Compute the curve point $(x_A, y_A) = k \times G$.
(4) Set $r := x_A + x \mod n$. If $r = 0$, go back to step 2.
(5) Set $s := k^{-1}(z + r d_A) \mod n$. If $s = 0$, go back to step 2.

Now (r, s) is a signature of m.

If $x = 0$, the algorithm above reduces to ECDSA []. Alice need not have any input into x per se, but Alice's signature loses deniability if x is discovered by an attacker.

2.2. **Verifying.** We assume that Bob knows Alice's public key Q_A, verified that Q_A is a valid curve point[], and that he knows both x and that x is random.

Bob verifies Alice's signature as follows.

(1) Check that $r, s \in [1, n-1]$. If not, the signature is invalid.

(2) Let z denote the leftmost l bits of hash(m) regarded as a number.
(3) Set $u_1 := zw \mod n$ and $u_2 := rw \mod n$ where $w := s^{-1} \mod n$.
(4) Compute the curve point $(x_B, y_B) = u_1 \times G + u_2 \times Q_A$.

The signature is valid if $r \equiv x_B + x \pmod{n}$, invalid otherwise.

Again if $x = 0$, the algorithm above reduces to ECDSA []. Anyone who can control x can forge the signature, so Bob must know that x is random.

We observe that the above algorithm requires only two scalar multiplications operations, making it faster thatn Triple Diffie-Hellman. In fact, these sums of two scalar multiplications can be computed even faster using Straus's algorithm aka Shamir's trick.[]

2.3. **Properties.** We prove that Alice and Bob construct the same curve point :

$$(x_B, y_B) = u_1 \times G + u_2 \times Q_A \qquad \text{by Bob 4} \tag{1}$$
$$= (zs^{-1} + rs^{-1}d_A)G \qquad \text{by Bob 3} \tag{2}$$
$$= (z + rd_A)s^{-1}G \tag{3}$$
$$= (z + rd_A)(z + rd_A)^{-1}kG \qquad \text{by Alice 5} \tag{4}$$
$$= (x_A, y_A) \qquad \text{by Alice 5} \tag{5}$$
(6)

It follows that a signed message will verify.

We must prove that forging a signature reguces to controlling x or violating Diffie-Hellman assumptions. We believe these arguments differ negligably from those for ECDSA, but there is no obvious reduction, so they'll need to be checked carefully.

3. Deniable EdDSA

In EdDSA, Alice sends two values (R, S) to Bob with R being a hash of part of Alice's private key with the message [, p. 6]. If Alice's private key is ever compromised, then these hashes alone provide a non-deniable hash based signature.

It's believeable that a signature scheme could be constructed using a different derivation of R from random data, partially supplied by Bob. We expect this changes the formal properties of EdDSA more radically than doing so with ECDSA did however, so perhaps a new name should be chosen.

E-mail address: `burdges@gnunet.org`

E-mail address: `grothoff@gnunet.org`