

## Re: DH Question

**Source:** <http://www.derkeiler.com/Newsgroups/sci.crypt/2004-06/2125.html>

---

**From:** Peter Fairbrother ([zenadsl6186\\_at\\_zen.co.uk](mailto:zenadsl6186_at_zen.co.uk))

**Date:** 06/29/04

Date: Tue, 29 Jun 2004 12:12:50 +0100

David Oxley wrote:

[The answers you wanted are: No. No benefit. Sign the whole.]

- > *It's not too difficult to see how a key exchange protocol like SKEME\**
- > *works in order to securely negotiate a session key with PFS*
- > *properties, but that relies on the ability to interact with the other*
- > *party everytime you need a new session key. But in times where the*
- > *latency of a given communication channel is extremely high, you want*
- > *the ability to come up with a key without having to talk to the other*
- > *party, and whilst remaining resistant to man-in-the-middle.*
- >
- > \* – <http://www.research.ibm.com/security/skeme.ps>

Non-interactive MITM resistance and FS? Tricky. The problem is getting Bob to delete a secret – after all Alice and Bob have to share a secret for there to be secure comms, and Bob has to know the secret.

Bob can provide a list of disposable keys, signed with a long-term key, and delete the secret keyparts when he gets a message. You then have to arrange for Alice and other users to get the keys, and not use a key for which Bob has deleted his part.

Bob can delete keyparts using a time-based schedule, giving FS after the key is deleted on a schedule, or he can deposit single-use keys on a server which allocates them as requested, giving FS after receipt.

m-o-o-t does both btw, in order to prevent a DoS due to single-use key exhaustion at the key server. It also uses preshared secrets and dummy traffic to give deniable-based-FS in the inbetween times. Only for your best friends :)

Nit: DH and SPEKE do not give perfect forward security. They do give forward security, but if an attacker can calculate discreet logs they are broke, so the forward security is not perfect. The only thing I know of that will give PFS is an otp.

--

Peter Fairbrother

Re: DH Question