Sketch of Security Proof for (n+1)Sec Protocol

The (n+1)Sec protocol is composed of following sub protocol:

1. **TDH**: Triple DH deniable Authentication
2. **FAGKE**: Flexible Authenticated Group Key Exchange protocol presented in [?]
3. **SecCom**: Secure (authenticated confidential) Send and Receive.
4. **TCA**: Transcript Consistency Assurance.

The threat model for each of these protocol is described in Section VI. The security of FAGKE is proven in the presented threat model. The SecComm consists of convential "sign" and "encrypt" functions and its security has been studied as a subprotocol to various protocols. We are not aware of any existing proof for TDH and TCA subprotocol.

The sketch of the proof goes as follows, in Section 1.2 and Section 2 we give convential formal proof of the security properties of TDH and TCA respectively. In Section 3 we reforumlates the proves of all four protocols in Protocol Composition Logic (PCL). In Section 4, we proof the security of (n+1)sec by proving the relative security of above sub prorotocol in relation to each other:

1. Q_1 as Parallel composition of TDH and FAGKE.
2. Sequential composition of Q_1 and SecCom.
3. Parallel compostion of SecCom and TCA.

1 Security of Triple Diffie-Hellman Authentication

1.1 The Triple Diffie-Hellman Protocol

Assuming that A and B are represeneted by long term public key g^A and g^B respectively:

Round 1	$A \rightarrow B$: $''A'', g^a$	$B \rightarrow A$:$'' B'', g^b$
Key Computation	$k \leftarrow H((g^b)^A \vert (g^B)^a \vert (g^b)^a)$	$k \leftarrow H((g^A)^b \vert (g^a)^B \vert (g^a)^b)$
Round 2	$\text{Enc}_k(H(k, A))$	$\text{Enc}_k(H(k, B))$

Table 1.

1.2 The deniablity of TDH

We will prove a parallel to Theorem 4 [?] which proves the deniability of SKEME. We use the notation which are introduced in Section ?. Following the same notation:

Definition 1. *By* $\text{Adv}^*_{\text{deny}}$ *we represent the party which represent the interaction of the Simulator* Sim *with the adverasy. In other word,* $\text{Adv}^*_{\text{deny}}$ *has access to all information which* Adv_{deny} *possess.*

Theorem 2. *If Computational Diffie-Hellman (CDH) is interactable then Triple DH Algorithm is deniable.*

Proof. We build Sim which interacts with Adv_{deny}. We show that if $\mathcal{J}$ is able to distinguish $\text{Trans}_{\text{Sim}}$ from $\text{Trans}_{\text{Real}}$, ze should be able to solve CDH as well.

Intuitively, when $\mathcal{A}_{\text{deny}}$ sends g^a to $\mathcal{S}_{\text{deny}}$, $\mathcal{S}_{\text{deny}}$ inquire $\mathcal{A}_{\text{deny}}$ for a, in this way $\mathcal{S}_{\text{deny}}$ also can compute the same key k by asking $\mathcal{A}^*_{\text{deny}}$. If $\mathcal{A}_{\text{deny}}$ has chosen $g^a \in \text{Tr}(B)$ or just chosen a random element of the group without knowing its DLP, then $\mathcal{S}_{\text{deny}}$ will choose a random exponent a' and computes the key k based on that and computes the confirmation value using k. Due to hardship of CDH this value is indistinguishable from a k generated by B

Now we suppose that the TDH is not deniable and we build a solver for CDH. First we note that if $\mathcal{A}_{\text{deny}}$ engages in an honest interaction with B there is no way that $\mathcal{J}$ can distinguish between the $T(\mathcal{A}_{\text{deny}}(\text{Aux}))$ and $T(\mathcal{S}_{\text{deny}}(\text{Aux}))$. As $\mathcal{A}_{\text{deny}}$ is able to generate the very exact transcript without help of B. Therefore, logically, the only possibility for $\mathcal{J}$ to distinguish $T(\mathcal{A}_{\text{deny}}(\text{Aux}))$ and $T(\mathcal{S}_{\text{deny}}(\text{Aux}))$ is when $\mathcal{A}_{\text{deny}}$ present $\mathcal{J}$ with a transcript that $\mathcal{A}_{\text{deny}}$ is not able to generate zirself. The only variable that $\mathcal{A}_{\text{deny}}$ has control over in the course of the exchange is g^a and therefore the only way $\mathcal{A}_{\text{deny}}$ is able to claim that ze were unable to generate the geneuine $T(\mathcal{A}_{\text{deny}}(\text{Aux}))$ is by submiting g^a which zirself does not know about its a exponent.

In such case, assuming the undeniability of TDH we have an ε such that

$$\max_{\text{all } \mathcal{J}} |2\Pr(\text{Output}(\mathcal{J}, \text{Aux}) = b) - 1| > \varepsilon$$

The solver $\mathcal{A}_{\text{CDH}}$ receives a triple (g, g^a, g^b) and should compute g^{ab}. To that end, assuming long term identiy g^A for some $\mathcal{A}_{\text{deny}}$, ze engages ,in a TDH key exchange with a hypothetical automated party $\mathcal{A}^*$ with long term private key B who generates g^b as the ephemeral key as well. $\mathcal{A}_{\text{CDH}}$, then toss a coin and based on the result it either choose a random a' and compute $g' = g^{a'}$ or set $g' = g^a$, then ze submits $h_0 = H\left(g^{bA}, g'^B, g^{ba'}\right)$ along side with (g^B, g^b) to the $\mathcal{J}$ as a proof of engagement with $\mathcal{A}^*$. Due to undeniability assumption

$$\text{Output}(\mathcal{J}, \text{Aux})(h_0, (A, g^a, B, g^b)) = b$$

with significant probablity as means $\mathcal{J}$ is able to distinguish $T(\mathcal{A}_{\text{deny}}(\text{Aux}))$ and $T(\mathcal{S}_{\text{deny}}(\text{Aux}))$ with high probablity. Therefore $\mathcal{J}$ is able to decide if:

$$h_0 \stackrel{?}{=\!=} H(g^{bA}, (g^a)^B, (g^a)^b)$$

Because H is a random oracle the only way that the judge is able to distinguish the second value from the real value is to have knowledge about the exact pre-image: $g^{bA}, (g^a)^B, (g^a)^b$. Using the information in the transcript $\mathcal{J}$ can compute $g^{bA}, (g^a)^B$, but still has to compute g^{ab} using g^a and g^b with high probablity without knowing a or b, at this point $\mathcal{A}_{\text{CDH}}$ is publishing the value of g^{ab}.

□

2 Security of Transcript Consistency Assurance

3 (n+1)Sec components in PCL Langugae

4 Security of composed sub protocols