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, Section 1 deals with security of TDH namely authentication and deniability. Section ? prove the security properties of the group key exchange protocol. Section 3 we give proof of the security properties of 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:

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

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

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

Intuitively, when $\mathcal{A}_{\mathrm{deny}}$ sends g^a to $\mathcal{S}_{\mathrm{deny}}$, $\mathcal{S}_{\mathrm{deny}}$ inquire $\mathcal{A}_{\mathrm{deny}}$ for a, in this way $\mathcal{S}_{\mathrm{deny}}$ also can compute the same key k by asking $\mathcal{A}^*_{\mathrm{deny}}$. If $\mathcal{A}_{\mathrm{deny}}$ has chosen $g^a \in \mathrm{Tr}(B)$ or just chosen a random element of the group without knowing its DLP, then $\mathcal{S}_{\mathrm{deny}}$ 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 $\mathcal{A}_{\mathrm{deny}}$ engages in an honest interaction with B there is no way that $\mathcal{J}$ can distinguish between the $T(\mathcal{A}_{\mathrm{deny}}(\mathrm{Aux}))$ and $T(\mathcal{S}_{\mathrm{deny}}(\mathrm{Aux}))$. As $\mathcal{A}_{\mathrm{deny}}$ is able to generate the very exact transcript without help of B. Therefore, logically, the only possibility for $\mathcal{J}$ to distinguish $T(\mathcal{A}_{\mathrm{deny}}(\mathrm{Aux}))$ and $T(\mathcal{S}_{\mathrm{deny}}(\mathrm{Aux}))$ is when $\mathcal{A}_{\mathrm{deny}}$ present $\mathcal{J}$ with a transcript that $\mathcal{A}_{\mathrm{deny}}$ is not able to generate zirself. The only variable that $\mathcal{A}_{\mathrm{deny}}$ has control over in the course of the exchange is g^a and therefore the only way $\mathcal{A}_{\mathrm{deny}}$ is able to claim that ze were unable to generate the geneuine $T(\mathcal{A}_{\mathrm{deny}}(\mathrm{Aux}))$ is by submiting g^a which zirself does not know about its a exponent.

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	$\mathrm{Enc}_k(H(k, A))$	$\mathrm{Enc}_k(H(k, B))$

Table 1.

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

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

The solver $\mathcal{A}_{\mathrm{CDH}}$ receives a triple (g, g^a, g^b) and should compute $g^{a b}$. To that end, assuming long term identiy g^A for some $\mathcal{A}_{\mathrm{deny}}$, 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. $\mathcal{A}_{\mathrm{CDH}}$, 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 $h_0 = H\big(g^{b\,A}, g'^B, g^{b\,a'}\big)$ along side with (g^B, g^b) to the $\mathcal{J}$ as a proof of engagement with $\mathcal{A}^*$. Due to undeniability assumption

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

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

$$h_0 \overset{?}{=} H(g^{b A}, (g^a)^B, (g^a)^b)$$

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: $g^{b A}$, $(g^a)^B$, $(g^a)^b$. Using the information in the transcript $\mathcal{J}$ can compute $g^{b A}, (g^a)^B$, but still has to compute g^{ab} using g^a and g^b with high probablity without knowing a or b, at this point $\mathcal{A}_{\mathrm{CDH}}$ is publishing the value of $g^{a b}$.

□

1.3 Confidentiality and Authenticity of TDH

In this section we prove that TDH is a secure two-party authenticated key exchange. We prove this in the model offered in [?].

2 Security of (n+1)sec authenticated group key exchange

In this section we prove the security of (n+1)sec group key exchange in the proposed adversarial model. Because the key exchange is essentially FAGKE with only difference is that the traditional DH key exchange replaced by TDH, we prove the security of (n+1)sec GKE based on the security of FAKE.

2.1 Security of GKE

We recall that the GKE protocol in (n+1)Sec is essentially the same as FAGKE protocol except that in (n+1)Sec we have:

$$k_{i, i+1} = H(g^{\mathrm{LS}_i x_{i+1}}, g^{\mathrm{LS}_{i+1} x_i}, g^{x_i x_{i+1}})$$

Where as in FAGKE we have:

$$k_{i, i+1} = g^{x_i x_{i+1}}$$

Therefore, to prove the that $(n+1)$Sec we need to prove Theorem 3:

Theorem 3. *If mBD+P protocol presented in [?] provides AKE-security of group keys, then so does the (n+1)sec key exchange.*

Proof. We only need to proof that if the adversary $\mathcal{A}_{\text{GKE}}$ can break the (n+1)sec with non-neglibile probablity then we can also break mBD+P protocol.

We note that any attack against $(n+1)$Sec can immediately re-interpreted as an attack to $\text{mBD}+P$ protots ocl, as long as it does not take advantage of the internal structure of k_{ij} values.

In particular games G0-G3 in proof of security of of mDB+P presented in [?] Theorem 4, is independent of how $k_{i,j}$ beside assuming that it exihibt a choice from random probablity distribution.

We modify G4 to adopt to $(n+1)$Sec protocol. A_{GKE} gets g^a and g^b from GDH challenge and embeds them as g^{x_i} and $g^{x_{i+1}}$. Ze generate random LS_i and LS_{i+1} and compute $k'_{i,i+1}$ using g^c value instead of $g^{x_i x_{i+1}}$ then it uses the Oracle to see if $k'_{i,i+1}$ is distinguishable from $k_{i,i+1}$ to solve the DDH problem. The remaining argument for game $G4$ is the same as $\text{mBD}+P$ proof. □

3 Security of Transcript Consistency Assurance