Static Diffie-Hellman with Keys of ECCSI Specified in RFC 6507

David Pointcheval[1] and Yang Yanjiang[2]

[1]DIENS, CNRS, ENS, PSL University, Paris, France
[2]Huawei, Singapore

February 3, 2018

Abstract

In this manuscript, we study the security of static Diffie-Hellman key exchange based on ECCSI (Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption) specified in RFC 6507, where the private keys are obtained essentially as Schnorr signatures (a variant).

1 Introduction

IETF RFC 6507 specifies an elliptic curve-based certificateless signature scheme, ECCSI (Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption). A Key Management Service (KMS) provides the root of trust for all users, and the global public key and secret key are called Public Authentication Key (KPAK) and Secret Authentication Key (KSAK), respectively. For each user, the key issued by KMS includes two components: one is Secret Signing Key (SSK) and the other is Public Validation Token (PVT).

This manuscript studies the security of static Diffie-Hellman key exchange using ECCSI's keys. Specifically, the static DH key exchange is described as follows.

ECCSI works in a subgroup $\mathbb{G}$ with prime order q, of an elliptic curve defined over F_p, where p is a prime. Let g be a generator of $\mathbb{G}$. For convenience of understanding, we denote arithmetic operations in $\mathbb{G}$'s as multiplicative. There are two hash functions: $\mathcal{H}$ onto the full domain $\mathbb{Z}_q$, and $\mathcal{H}'$ onto the ℓ-bit strings, to generate the session keys with the static Diffie-Hellman key exchange.

- The setup algorithm of ECCSI is: the KMS chooses a random secret non-zero element $x \xleftarrow{R} \mathbb{Z}_q^*$, and sets $X \leftarrow g^x$. The latter is the Public Authentication Key, $\mathsf{KPAK} = X$ and the former is the Secret Authentication Key, $\mathsf{KSAK} = x$;

- The key issuance algorithm of ECCSI works as follows: for a new identity id, the KMS chooses a random non-zero element $r \xleftarrow{R} \mathbb{Z}_q^*$, computes $R \leftarrow g^r$ and sets $s \leftarrow x + r \cdot \mathcal{H}(g, X, \mathsf{id}, R) \bmod q$. The PVT for id is $\mathsf{pvt}_{\mathsf{id}} \leftarrow R$, while the SSK is $\mathsf{ssk}_{\mathsf{id}} \leftarrow s$. It can be seen that (R, s) is a variant of Schnorr Signature;

- In order to agree on a common session key with id', the user with identity id will compute $v = (X \cdot \mathsf{pvt}_{\mathsf{id}'}^{\mathcal{H}(g,X,\mathsf{id}',\mathsf{pvt}_{\mathsf{id}'})})^{\mathsf{ssk}_{\mathsf{id}}}$ and $K = \mathcal{H}'(\mathsf{id}, \mathsf{id}', v)$. His partner can do the same as $v' = (X \cdot \mathsf{pvt}_{\mathsf{id}}^{\mathcal{H}(g,X,\mathsf{id},\mathsf{pvt}_{\mathsf{id}})})^{\mathsf{ssk}_{\mathsf{id}'}}$ and $K' = \mathcal{H}'(\mathsf{id}, \mathsf{id}', v')$:

$$\begin{aligned} v = (X \cdot \mathsf{pvt}_{\mathsf{id}'}^{\mathcal{H}(g,X,\mathsf{id}',\mathsf{pvt}_{\mathsf{id}'})})^{\mathsf{ssk}_{\mathsf{id}}} &= (g^{x+r' \cdot \mathcal{H}(g,X,\mathsf{id}',\mathsf{pvt}_{\mathsf{id}'})})^{\mathsf{ssk}_{\mathsf{id}}} \\ &= g^{\mathsf{ssk}_{\mathsf{id}'}\mathsf{ssk}_{\mathsf{id}}} = (X \cdot \mathsf{pvt}_{\mathsf{id}}^{\mathcal{H}(g,X,\mathsf{id},\mathsf{pk}_{\mathsf{id}})})^{\mathsf{ssk}_{\mathsf{id}'}} = v' \end{aligned}$$

One can note that $v = v' = \mathsf{DH}(X \cdot \mathsf{pvt}_{\mathsf{id}}^{\mathcal{H}(g,X,\mathsf{id},\mathsf{pvt}_{\mathsf{id}})}, X \cdot \mathsf{pvt}_{\mathsf{id}'}^{\mathcal{H}(g,X,\mathsf{id}',\mathsf{pvt}_{\mathsf{id}'})})$.

2 Security Analysis

We consider a security model where there is a target user $\mathsf{id}^\#$ pre-determined, with keys $(\mathsf{ssk}^\#, \mathsf{pvt}^\#)$, and the adversary would like to generate, with $\mathsf{id}^\#$, a public validation token pvt^* for some identity id^* together with the common key: given $(\mathsf{id}^\#, R^\# = \mathsf{pvt}^\#)$, it can output a new public validation token pvt^* so that it can distinguish the real key K from a random one, with possible access to the KMS for many identities. We just want id^* to be a new identity. Trivially, the adversary is not allowed to get the secret signing key corresponding to $(\mathsf{id}^*, \mathsf{pvt}^*)$.

The semantic security of K means that no adversary is able to generate such a new pvt^* for which it can distinguish the real session key from a random session key.

Note that we are in the random oracle model, and so, the only way to distinguish K is to query v to the random oracle $\mathcal{H}'$, and so we can extract $\mathsf{DH}(X \cdot (\mathsf{pvt}^\#)^{\mathcal{H}(g,X,\mathsf{id}^\#,\mathsf{pvt}^\#)}, X \cdot (\mathsf{pvt}^*)^{\mathcal{H}(g,X,\mathsf{id}^*,\mathsf{pvt}^*)})$ from the list of $\mathcal{H}'$-queries. Hence, we can hope to make the security rely on the CDH assumption: for a bias ε on this guess, the Diffie-Hellman value must be in the list with probability ε.

More precisely, we can prove the following security result:

Theorem 2.1 (Semantic Security of the Static ECCSI-Authenticated Diffie-Hellman Key Exchange)**.** *If an adversary can break the semantic security of the key K within time t and advantage ε, then one can break the CDH problem within time $4t$ and probability $\varepsilon^4/16Q^6Q'^4$, where Q and Q' are the number of queries to the random oracles $\mathcal{H}$ and $\mathcal{H}'$ respectively.*

Proof. As said above, the proof holds in the random oracle model, so we consider the lists Λ and Λ' of all the query-answer pairs for $\mathcal{H}$ and $\mathcal{H}'$ respectively. We first show how to compute $g^{x^2} = \mathsf{DH}(X, X)$, given $X = g^x$.

- Set X to be the KPAK.
- Let us first simulate the target public key $\mathsf{pvt}^\# = R^\#$ for $\mathsf{id}^\#$: $R^\# \leftarrow X^{\rho^\#}$, for $\rho^\# \stackrel{R}{\leftarrow} \mathbb{Z}_q^*$.
- Let us now simulate key queries: for an identity id, the simulator chooses a random $s \stackrel{R}{\leftarrow} \mathbb{Z}_q$ and a random $h \stackrel{R}{\leftarrow} \mathbb{Z}_q$, it then sets $R \leftarrow g^{s/h} X^{-1/h}$, as well as $\mathcal{H}(g, X, \mathsf{id}, R) \leftarrow h$. The simulation is perfect, unless the query (g, X, id, R) has already been asked to $\mathcal{H}$ which is quite unlikely since R is uniformly distributed (we ignore this bad case).
- Let us now assume that the adversary outputs $(\mathsf{id}^*, \mathsf{pvt}^*)$ so that he can later distinguish the real key K from a random one. This means that the adversary outputs $(\mathsf{id}^*, \mathsf{pvt}^*)$, and asks the query $\mathsf{DH}(X \cdot (\mathsf{pvt}^\#)^{\mathcal{H}(g,X,\mathsf{id}^\#,\mathsf{pvt}^\#)}, X \cdot (\mathsf{pvt}^*)^{\mathcal{H}(g,X,\mathsf{id}^*,\mathsf{pvt}^*)})$ to the random oracle $\mathcal{H}$: by choosing it at random, with probability $1/Q'$, we extract the correct value, where Q' is the number of queries to $\mathcal{H}'$. Hence, our extractor succeeds with probability ε/Q'.

  We can note $H^\# = \mathcal{H}(g, X, \mathsf{id}^\#, \mathsf{pvt}^\#)$ and $H^* = \mathcal{H}(g, X, \mathsf{id}^*, \mathsf{pvt}^*)$. Among the Q $\mathcal{H}$-queries, there is an index I for which the success probability with H^* corresponding to the I-th $\mathcal{H}$-query is at least ε/QQ'. We choose this index at random, and with probability greater than $1/Q$, this is the good choice. Then, using the Forking Lemma, with probability $\varepsilon/2QQ'$ we have a success with a good beginning, and then just replaying the same execution with a new random value H' for $\mathcal{H}(g, X, \mathsf{id}^*, \mathsf{pk}^*)$, we obtain a second success with probability $\varepsilon/2QQ'$. Hence, with probability

$\varepsilon^2/4Q^3{Q'}^2$ and 2 executions of the attack, we obtain pvt^*, H^* and H' such that

$$\begin{aligned} v = \mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X \cdot (\mathsf{pvt}^*)^{H^*}) &= \mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X) \cdot \mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X^{\rho^* \cdot H^*}) \\ &= \mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X) \cdot \mathsf{DH}(X^{1+\rho^{\#} \cdot H^{\#}}, X^{\rho^* \cdot H^*}) \\ v' = \mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X \cdot (\mathsf{pvt}^*)^{H'}) &= \mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X) \cdot \mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X^{\rho^* \cdot H'}) \\ &= \mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X) \cdot \mathsf{DH}(X^{1+\rho^{\#} \cdot H^{\#}}, X^{\rho^* \cdot H'}) \end{aligned}$$

By computing $\frac{v^{H'}}{v'^{H^*}}$, we obtain

$$\frac{v^{H'}}{v'^{H^*}} = \frac{\mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X)^{H'}}{\mathsf{DH}(X \cdot (\mathsf{pvt}^{\#})^{H^{\#}}, X)^{H^*}} = \mathsf{DH}(X^{1+\rho^{\#} \cdot H^{\#}}, X^{H'-H^*}) = \mathsf{DH}(X, X)^{(1+\rho^{\#} \cdot H^{\#})(H'-H^*)}.$$

This leads to $\mathsf{DH}(X, X)$.

Next, one can note that from an algorithm $\mathcal{A}$ that solves the Square Diffie-Hellman problem with probability $\hat{\varepsilon}$ within time $\hat{t}$, one solves the usual Diffie-Hellman problem: given $X = g^x$ and $Y = g^y$, one runs $\mathcal{A}$ on XY, and gets $g^{(x+y)^2}$ with probability $\hat{\varepsilon}$; one then runs $\mathcal{A}$ on X/Y, and gets $g^{(x-y)^2}$ with probability $\hat{\varepsilon}$. Hence, with probability $\hat{\varepsilon}^2$, within time $2\hat{t}$, one can compute $(g^{(x+y)^2}/g^{(x-y)^2})^{1/4} = g^{xy}$. □