*2. In Subsection 5.2, the description of ABE components is unclear:*

*(1) typo of redundant double prime('') in PK,*

**Correct the following typos (yellow)**

Public key (PK) depends on the master key values and is kept in clear allowing users to access the information: PK = (g t1 , gt2 ”, · · · , gtn , e(g, g) y ). Here e(g, g) is the bilinear pairing function corresponding to an elliptic curve

*(2) no description of basic properties of bilinear paring function,*

**I suggest to insert text of the following form.**



*(***this has to referenced because this is taken from work [13 in list])**

*(3) no description of the meanings of the notations w and s.*

**I suggest the following modification**

Secret user KEY SET depends on his attribute set. Here each Di (GROUP KEY) serves for decryption of the data of a single group of users, for example, related to some project: {ti}U → D = {Di = g yw ti }. Here w \in Z is the unique security parameter allowing to control the key generation for each user group.

Encrypted text M, in this context, M = F ILE KEY , or the permanent AES symmetric key, which allows to avoid the file re-encryption. The encryption procedure is performed by a multiplication. The set of the public keys Ei (PUBLIC SHARE KEY) corresponding to the set of groups able to access the text is kept along with the encrypted text E: E = Me(g, g) ys , {Ei = g tis w }, ∀i ∈ {ti}M. Here s \in Z is the unique security parameter allowing to control the key generation for each file share.

*(4) bad-form notation "for all i in {t\_i}\_M" (because "i" cannot belong to the set).*

Encrypted text M, in this context, M = F ILE KEY , or the permanent AES symmetric key, which allows to avoid the file re-encryption. The encryption procedure is performed by a multiplication. The set of the public keys Ei (PUBLIC SHARE KEY) corresponding to the set of groups able to access the text is kept along with the encrypted text E: E = Me(g, g) ys , {Ei = g tis w }, ∀i, t\_i ∈ {t\_i}\_U.

*3. In Subsection 5.3, the description of SSS (modular secret sharing) is unclear:*

*(1) Di is a typo,*

**I suggest the following modification**

The attribute-based private keys D={D\_i} should be protected while being stored in the device memory

*(2) the notation based on Chinese Remainder Theory (CRT) is not well presented,*

KEY SET KEY is a secret value and it is split by the secure method of polynomial modular secret sharing [11, 10] into the set of 4 shares: KEY SET KEY = PASS + PIN + TIME + DEV PASS. The adversary cannot get any information of the KEY SET KEY unless he possesses all 4 key parts. The PASS and PIN values are predefined, similar to the key storing construction based on the polynomial secret sharing scheme (SSS) in [11].

In this SSS, the following notation is used:

S(x) \in F\_2[x] denotes the secret polynomial with coefficients from the finite field F\_2 to be computed by pooling the shares together.

s\_1(x), s\_2(x) \in F\_2[x] denote the shares of the SSS, that are used to compute the secret s(x)

The proposed authentication system is based on the shared storing of the user key. Also, the mobile client acts as a dealer in the SSS. Using the SSS ensures that the key can only be accessed by an authenticated user. The participants of the (2, 2)-threshold SSS are the user and device. The user share s1(x) is computed based on the PIN and the PASS entered by the user. Additionally, the current time value TIME is used in the calculation of the share. Let s(x) = d and s1(x) = f(P IN + P ASS + T IME), where f is a one-way (hash) function that transforms the data into the string of the desired length:

*(3) the meanings of d, S, p\_0, and p\_1 are not clear,*

*(4) What are the reasons for S being equivalent to s and s\_1 in modulo p\_0 and p\_1, respectively?*

S = s mod p0 S = s1 mod p1 (2) According to the CRT: S ≡ s1p0p0 −1 p1 + s1p1p1 −1 p0 + mod p0p1 (

Here

P0 and p1 \in F\_2[x] denote the public moduli of the polynomial SSS [10, 11] and s \in F\_2[x] denotes the intermediate secret value. The formulae for these moduli, secret and participants’ shares directly follow from the definition of polynomial SSS construction.

*2. More discussions on the figure 1 and 2 should be added. The current version is not easy to understand.*

*Figure 1 describes the mobile client protection both in online and offline mode. In online mode, the client has the possibility to connect to the server and the security of the client is enhanced by the server-backed up mechanisms. On the other hand, in offline mode the client’s security is supported by the standalone mechanisms. Additionally, the mobile client protection is enhanced by the threat intelligence unit providing the constant monitoring and analysis.*

*Figure 2 depicts the client-side protection mechanisms. The client should support 4 subsystems: i) encryption subsystem that contains the procedures of encryption and decryption; ii) storage subsystem that provides the downloaded shares and key storage protection; iii) threat intelligence unit that provides the constant monitoring, and iv) the communication subsystem. In short, all security procedures are connected to 4 groups of operations: file request and receiving; encryption and decryption; file and key storing; monitoring and analysis.*

*3. The figure 8 should be explained more in detail.*

As it is possible to be observed in Figure 8, the master key on the upper level is computed by means of SSS (section 5.3). It serves for decrypting the ABE key storage (section 5.2). The ABE key allows the user finally decrypt the permanent file share keys. Further, the file share can be decrypted by means of AES.

**THIS IS A CORRECTION:**

**The levels of figure 8 should contain the following text:**

**----Master key (AES)= device key+ timestamp + user password + user PIN**

**----Group ABE key = private key + public key**

**----****AES file share key generated by server**