Bernd Zwattendorfer, Daniel Slamanig - "Privacy-Preserving Realization of the STORK Framework in the Public Cloud" - 10th International Conference on Security and Cryptography (SECRYPT 2013), 2013, pp. 419-426
http://www.secrypt.icete.org/?y=2013

Privacy-Preserving Realization of the STORK Framework in the Public Cloud

Bernd Zwattendorfer and Daniel Slamanig
Institute for Applied Information Processing and Communications (IAIK)
Graz University of Technology (TUG), Inffeldgasse 16a, 8010 Graz, Austria
{bernd.zwattendorfer,daniel.slamanig}@iaik.tugraz.at

Keywords: STORK, PEPS, Public Cloud, eID, eID Federation, Privacy-Preservation, Proxy Re-Encryption

Abstract: The STORK framework – enabling secure eID federation across European countries – will be the dominant identification and authentication framework across Europe in the future. While still in its start up phase, adoption of the STORK framework is continuously increasing and high loads can be expected, since, theoretically, the entire population of the European Union will be able to run authentications through this framework. This can easily lead to scalability issues, especially for the proxy-based (PEPS) approach in STORK, which relies on a central gateway being responsible for managing and handling citizen authentications. In order to mitigate the associated scalability issues, the PEPS approach could be moved into the public cloud. However, a move of a trusted service into the public cloud brings up new obstacles, especially with respect to citizens' privacy. In this paper we propose an approach how this move could be successfully realized by still preserving citizens' privacy and keeping existing national eID infrastructures untouched. We present the approach in detail and evaluate its capability with respect to citizens' privacy protection as well as its practicability. We conclude, that the proposed approach is a viable way of realizing an efficient and scalable Pan-European citizen identification and authentication framework.

1 INTRODUCTION

To provide secure and reliable e-Government services to their citizens, many European countries have already rolled out secure eID solutions. While such national eID solutions have been deployed in the field for many years, they still have been lacking cross-border acceptance. To overcome this lack in interoperability, the European Commission had launched the large scale pilot (LSP) project STORK[1] (*Secure Identity Across Borders Linked*), which involved 18 EU Member States. The developed STORK framework is currently heavily pushed by the European Commission (EC) and will be the dominant identification and authentication framework across Europe in the future. To achieve improved maturity and sustainability of this framework, the successor project STORK 2.0[2] currently aims on further piloting.

Basically, the STORK framework is based on two different interoperability models, the *Middleware* (MW) model and the *Pan-European Proxy Service* (PEPS) model. In this paper, we focus on the PEPS model only. The PEPS model sets up on a centralized approach, where a central single gateway (PEPS) per member state acts as intermediary (gateway) between the service providers and the national eID framework. This PEPS is operated in a traditional data center within a trusted environment. However, the use of STORK is continuously increasing and high load can be expected in the future. This may induce scalability problems, especially for the PEPS approach, as theoretically the population of an entire country can use the PEPS for authentications.

In this paper we propose an advanced STORK framework by moving the centralized PEPS instances into the public cloud. Scalability problems are mitigated due to high elasticity and scalability provided by the public cloud. However, other obstacles particularly regarding to the citizens' privacy emerge. We describe how such a move could successfully be realized by preserving citizens' privacy and by still keeping the existing infrastructure nearly untouched. We thereby assume, as it is common, that the cloud providers are *honest but curious*, meaning that they perform their computing tasks correctly, but may try to extract as much information as possible.

[1] https://www.eid-stork.eu
[2] https://www.eid-stork2.eu

2 CRYPTOGRAPHIC BUILDING BLOCKS

In this section we briefly discuss required cryptographic primitives. Note, that in practice in all discussed schemes the public key pk_A of a user A is bound to the user's identity. This is typically realized by means of digital certificates within some public key infrastructure (PKI). Thus, we assume public keys always being publicly available in an authentic fashion. Furthermore, if an entity A is in possession of an encryption (PKE) and signature (DSS) key pair, we denote them by $(\mathsf{sk}_\mathsf{A}, \mathsf{pk}_\mathsf{A})$ and $(\mathsf{sk}'_\mathsf{A}, \mathsf{pk}'_\mathsf{A})$ respectively. Moreover, we assume that if bitstrings a and b are concatenated $a\|b$, this happens in a way such that all individual components are uniquely recoverable. When we want to have authenticated encryption in the public key setting, we typically apply the generic sign-then-encrypt strategy, but sometimes need to rely on the generic encrypt-then-sign strategy (An, 2001).

2.1 Digital Signatures

A digital signature scheme (DSS) is a triple $(\mathcal{K}, \mathcal{S}, \mathcal{V})$ of poly-time algorithms, whereas $\mathcal{K}$ is a probabilistic key generation algorithm that takes a security parameter κ and outputs a private and public key pair $(\mathsf{sk}, \mathsf{pk})$. The probabilistic signing algorithm $\mathcal{S}$ takes as input a message $M \in \{0,1\}^*$ and a private key sk, and outputs a signature σ. The verification algorithm $\mathcal{V}$ takes as input a signature σ, a message $M \in \{0,1\}^*$ and a public key pk, and outputs a single bit $b \in \{\mathtt{true}, \mathtt{false}\}$ indicating whether σ is a valid signature for M. Furthermore, we require the DSS to be correct, i.e., for all $(\mathsf{sk}, \mathsf{pk}) \in \mathcal{K}(\kappa)$ and all $M \in \{0,1\}^*$ we have $\mathcal{V}(\mathcal{S}(M, \mathsf{sk}), M, \mathsf{pk}) = \mathtt{true}$. A DSS is secure, if it is existentially unforgeable under adaptively chosen-message attacks (UF-CMA). We note that in practice one typically employs the hash-then-sign paradigm, i.e., instead of inputting M into $\mathcal{S}$ and $\mathcal{V}$, one inputs $H(M)$ where H is a suitable cryptographic hash function.

2.2 (Public Key) Encryption

A public key encryption (PKE) scheme is a triple $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ of poly-time algorithms, whereas $\mathcal{K}$ is a probabilistic key generation algorithm that takes a security parameter κ and outputs a private and public key pair $(\mathsf{sk}, \mathsf{pk})$. The probabilistic encryption algorithm $\mathcal{E}$ takes as input a public key pk and a message $M \in \{0,1\}^*$ and returns a ciphertext $c = \mathcal{E}(\mathsf{pk}, M)$. The decryption algorithm $\mathcal{D}$ takes as input a private key sk and a ciphertext c and returns a message $M = \mathcal{D}(\mathsf{sk}, c)$ or $\perp$ in the case of failure. We require a PKE scheme to be indistinguishable under chosen plaintext attacks (IND-CPA). Abstractly, we can define private key (or symmetric) encryption schemes analogously, whereas $\mathcal{K}$ only generates a single key K which is used as input to the encryption and decryption algorithms. For the security of a private key encryption scheme we also require IND-CPA security. Note that when we speak of applying PKE to a larger message M, then we implicitly mean applying *hybrid encryption*, i.e., choosing a random K and sending/storing the tuple $(c_1 = \mathcal{E}(K, \mathsf{pk}), c_2 = \mathcal{E}'(M, \mathsf{K}))$.

2.3 Proxy Re-Encryption

An unidirectional multi-use proxy re-encryption (UM-PRE) scheme allows a semi-trusted proxy to transform a message encrypted under the key of party A into another ciphertext, containing the initial plaintext, such that another party B can decrypt the ciphertext with its key. The proxy neither gets access to the plaintext nor to the decryption keys. The proxy can only transform in one direction and one ciphertext can be transformed multiple times. Therefore, one may either use suitable identity-based (Green and Ateniese, 2007) or a traditional (Chow et al., 2010) multi-use proxy re-encryption scheme. An UM-PRE is a tuple $(\mathcal{S}, \mathcal{K}, \mathcal{RK}, \mathcal{E}_\mathcal{R}, \mathcal{RE}, \mathcal{D}_\mathcal{R})$ of poly-time algorithms. The algorithm $\mathcal{S}$ runs a setup and produces system parameters params. $\mathcal{K}$ is a probabilistic key generation algorithm that takes a security parameter κ and outputs a private and public key pair $(\mathsf{rsk_i}, \mathsf{rpk_i})$. The re-encryption key generation algorithm $\mathcal{RK}$ takes as input a private key $\mathsf{rsk_i}$ and another public key $\mathsf{rpk_j}$ and outputs a re-encryption key $\mathsf{rk_{i\to j}}$. The probabilistic encryption algorithm $\mathcal{E}_\mathcal{R}$ gets a public key $\mathsf{rpk_i}$ and a plaintext M and outputs $c_i = \mathcal{E}_\mathcal{R}(\mathsf{rpk_i}, M)$. The (probabilistic) re-encryption algorithm gets as input a ciphertext c_i under $\mathsf{rpk_i}$ and a re-encryption key $\mathsf{rk_{i\to j}}$ and outputs a re-encrypted ciphertext $c_j = \mathcal{RE}(\mathsf{rk}_{i\to j}, c_i)$ for $\mathsf{rpk_j}$. The decryption algorithm $\mathcal{D}_\mathcal{R}$ takes private key $\mathsf{rsk_j}$ and a ciphertext c_j and outputs $M = \mathcal{D}_\mathcal{R}(\mathsf{rsk_j}, c_j)$ or an error $\perp$. Here, we require indistinguishability under chosen ciphertext attacks (IND-CCA).

3 THE STORK FRAMEWORK AND PEPS MODEL

The general objective of STORK was to tackle the issue of eID heterogeneity across Europe and to achieve cross-border acceptance of eIDs. To achieve that, STORK developed an interoperability layer on

top of the existing national solutions to avoid any severe changes in the individual national infrastructures. By the help of STORK, EU citizens are now able to securely authenticate at foreign online applications by using their own national eID issued by their home country. To make the individual national eID solutions comparable, STORK quantitatively modeled the strength of the different authentication mechanisms and quality of eID registration to STORK defined quality authentication assurance (QAA) levels.

Basically, the STORK framework sets up on two different identity models. The first model is called middleware (MW), where users directly authenticate at the service provider. The second model relies on a proxy-based approach, where a proxy (PEPS) acts as intermediary (gateway) between the user and the service provider. In this paper, we focus on the PEPS model only. Further details on both models can be found in (Leitold and Zwattendorfer, 2010; Zwattendorfer et al., 2013). The PEPS model, its setup, and its detailed process flow will be presented in the next subsections.

3.1 PEPS Model

In this model, a proxy or gateway (PEPS) acts as intermediary between the service provider and the actual authenticating entity (identity provider). Hence, segmented trust relationships apply. The PEPS basically has three functionalities. First, the PEPS is hiding the complexity and specifics of national eID solutions to other countries. Thereby, the PEPS connects to all required existing national identity providers (IdPs) or attribute providers (APs). Second, it enables the secure transfer of identity data to service providers. Different service providers can connect to the PEPS using appropriate interfaces. Third, the PEPS implements the STORK protocol (STORK, 2011b), which is based on the Security Assertion Markup Language (SAML)[3] and allows for secure cross-border identification and authentication. Figure 1 illustrates the PEPS model in a cross-border identification and authentication scenario.

More precisely, a PEPS can either act as S-PEPS (PEPS in the service provider's home country) or as C-PEPS (PEPS in the citizen's home country). The S-PEPS communicates with different service providers and acts as intermediary between the service providers and the C-PEPS. The C-PEPS triggers identification and authentication of a citizen at an IdP and/or AP and thereby acts as intermediary between an S-PEPS and the IdP and/or AP. A basic cross-border PEPS-PEPS authentication scenario is illustrated in Figure 1. Subsequently, we present details on the required setup and the process flow in a cross-border PEPS-PEPS scenario (cf. Figure 2).

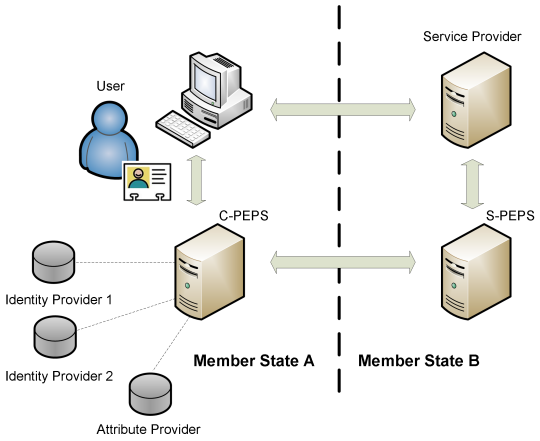

Figure 1: PEPS Model

3.1.1 Setup

Within the STORK project, a cross-border interoperability protocol (STORK, 2011b) for a secure identity and authentication data exchange had been developed. This protocol is mainly applied in the cross-border context, involving PEPS and MW components. In the STORK PEPS-PEPS model, the STORK protocol is used in the communication between an S-PEPS and a C-PEPS and vice versa.

For setting up a STORK communication using its protocol, appropriate metadata information has to be exchanged between the involved STORK entities. In particular, for the PEPS-PEPS scenario these metadata includes the endpoint location URL of every PEPS ($\mathsf{URL}_{\mathsf{S-PEPS}}$ and $\mathsf{URL}_{\mathsf{C-PEPS}}$), where messages should be delivered to, and its signature certificate (including the PEPS public key $\mathsf{pk}'_{\mathsf{S-PEPS}}$ or $\mathsf{pk}'_{\mathsf{C-PEPS}}$). The exchanged STORK messages between the individual PEPS will be digitally signed using the corresponding private signing key ($\mathsf{sk}'_{\mathsf{S-PEPS}}$ or $\mathsf{sk}'_{\mathsf{C-PEPS}}$) to ensure integrity and authenticity of the exchanged messages. Currently, PEPS metadata exchange is handled by the individual countries using various organizational means. However, the STORK 2.0 consortium together with the European Commission are currently in the progress of setting up an appropriate governance structure for managing these metadata, where a European Commission body will be responsible for PEPS metadata management and maintenance.

Basically, to accomplish a PEPS-PEPS authentication scenario, the following setup is required. Every PEPS country generates an individual DSS key pair for their PEPS, $(\mathsf{sk}'_{\mathsf{S-PEPS_i}}, \mathsf{pk}'_{\mathsf{S-PEPS_i}})$

[3] http://saml.xml.org

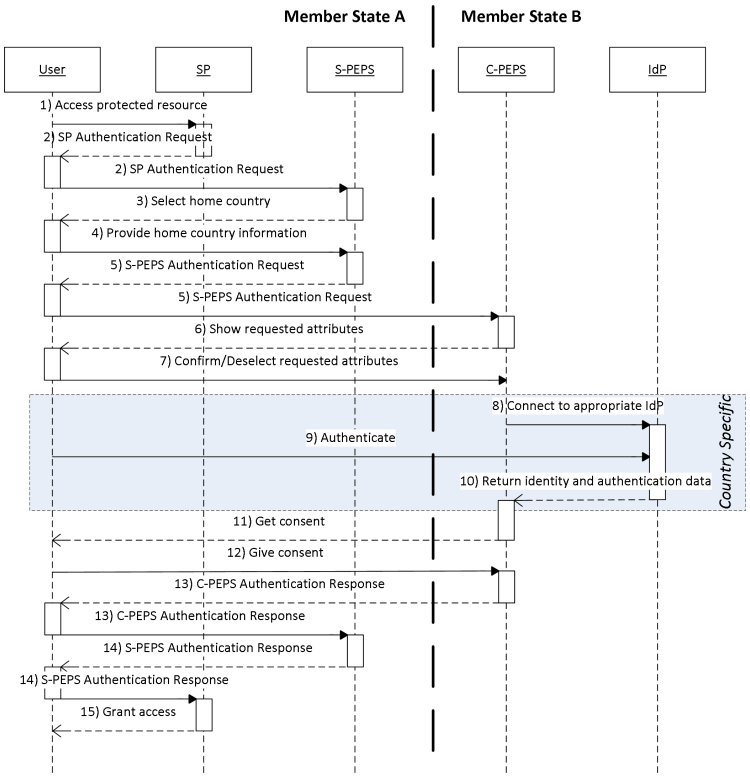

Figure 2: The PEPS-PEPS Process Flow

and $(\mathsf{sk}'_{\mathsf{C-PEPS_i}}, \mathsf{pk}'_{\mathsf{C-PEPS_i}})$ respectively. Both $\mathsf{pk}'_{\mathsf{S-PEPS_i}}$ and $\mathsf{pk}'_{\mathsf{C-PEPS_i}}$ as well as $\mathsf{URL}_{\mathsf{S-PEPS_i}}$ and $\mathsf{URL}_{\mathsf{C-PEPS_i}}$ are published. Additionally, each service provider SP_j relying on PEPS authentication creates a DSS key pair $(\mathsf{sk}'_{\mathsf{SP_j}}, \mathsf{pk}'_{\mathsf{SP_j}})$ to sign its messages. Of course, a setup procedure of the IdP is necessary too. However, we consider this procedure country-specific and hence do not present any details.

Since, instead of the SP, the individual PEPS as trusted intermediaries are liable for any fraudulent activity, each PEPS is responsible to log certain authentication and transaction information in case a later audit or inspection is required. These log data include information on the sender for incoming messages, on the receiver for outgoing messages, a message identifier, date/time of the message, a hash value of the message (using SHA-256), and a log entry counter (STORK, 2011a).

3.1.2 Process Flow

The detailed cross-border PEPS-PEPS process flow as illustrated in Figure 2 will be the basis to discuss our cloud-based PEPS-PEPS approach. For the sake of simplicity, we denote the particular entities $S-PEPS_i$, $C-PEPS_i$, SP_j and IdP_k in the process flow simply by SP, $S-PEPS$, etc.

1) Access protected resource: A citizen (user) originating from country B wants to access a protected (e-Government) service in the foreign country A.

2) SP authentication request: The SP has no security context established with the user yet and thus requires authentication at a specific strength `reqQAA`. Furthermore, specific citizen attributes `reqAttr` are required in order to provide the requested service to the user. The authentication request of the SP including `reqAttr`, `reqQAA`, and the name of the SP `sp_name` is signed by the SP resulting in $\sigma_{SP} = \mathcal{S}(\mathsf{sk}'_{\mathsf{SP}}, \mathtt{reqAttr} \| \mathtt{reqQAA} \| \mathtt{sp\_name})$, and the tuple $(\mathtt{reqAttr}, \mathtt{reqQAA}, \mathtt{sp\_name}, \sigma_{SP})$ is forwarded via the user to the S-PEPS. The protocol for transmitting the SP authentication request can be nation-specific.

3) Select home country: The S-PEPS verifies σ_{SP} and if valid, the S-PEPS provides the citizen a country selection page.

4) Provide home country information: The citizen selects her country of origin and submits the information `country` (B) back to the S-PEPS.

5) S-PEPS authentication request: The S-PEPS

takes the information received from the SP ($\texttt{reqAttr}, \texttt{reqQAA}, \texttt{sp\_name}$) and signs it using its private key sk'_{S-PEPS} resulting in $\sigma_{S-PEPS} = \mathcal{S}(\mathsf{sk}'_{\mathsf{S-PEPS}}, \texttt{reqAttr}\|\texttt{reqQAA}\|\texttt{sp\_name})$. Depending on country, the S-PEPS forwards its authentication request including the data ($\texttt{reqAttr}, \texttt{reqQAA}, \texttt{sp\_name}, \sigma_{S-PEPS}$) to the C-PEPS, which is responsible for authentications of country, i.e., $C-PEPS$ of country B.

6) Show requested attributes: The C-PEPS verifies σ_{S-PEPS} and sends reqAttr and sp_name for attribute selection to the citizen.

7) Confirm/deselect requested attributes: According to STORK, citizens must confirm the reception of reqAttr from the IdP. Thereby, citizens can deselect attributes for denying further transmission of those attributes to the SP, whereas only attributes which have been requested as being optional by the SP can be deselected.

8) Connect to appropriate IdP: The implementation of this step and the subsequent steps 9 and 10 are country specific. Depending on reqQAA, the appropriate IdP for citizen identification and authentication is contacted. The C-PEPS forwards the requested attributes reqAttr to the IdP.

9) Authenticate: Depending on the IdP, the citizen authenticates using the respective authentication method. For the highest authentication level reqQAA, the citizen must authenticate using her official national eID.

10) Return identity and authentication data: The IdP returns citizen's identity and attribute information attr, which include the attribute values corresponding to reqAttr, to the C-PEPS. Additionally, QAA stating the actual level of the authentication process is returned.

11) Get consent: The C-PEPS sends attr and sp_name to the citizen to get consent for further transmission of attr to the C-PEPS and then subsequently to SP.

12) Give consent: The citizen may give or deny consent for attr transmission.[4]

13) C-PEPS authentication response: The C-PEPS takes attr and QAA, signs it generating $\sigma_{C-PEPS} = \mathcal{S}(\mathsf{sk}'_{\mathsf{C-PEPS}}, \texttt{attr}\|\texttt{QAA})$, and transfers ($\texttt{attr}, \texttt{QAA}, \sigma_{C-PEPS}$) to the S-PEPS.

14) S-PEPS authentication response: The S-PEPS verifies σ_{C-PEPS}, signs again attr and QAA resulting in $\sigma'_{S-PEPS} = \mathcal{S}(\mathsf{sk}'_{\mathsf{S-PEPS}}, \texttt{attr}\|\texttt{QAA})$, and transfers ($\texttt{attr}, \texttt{QAA}, \sigma_{S-PEPS}$) to the SP.

15) Grant access: The SP verifies σ_{S-PEPS} and checks if the citizen has been authenticated with the requested reqQAA. Based on the attributes attr received, the SP either grants or denies access to the requested service to the citizen.

[4]Note, that depending on the country and its implementation, Steps 6 and 7 as well as Steps 10 and 11 can also be included in Step 9.

4 PORTING THE PEPS MODEL INTO THE PUBLIC CLOUD

The current PEPS Model relies on a central PEPS instance, which is operated for every single country in a trusted conventional data center. Since STORK will be the dominant authentication framework across Europe in the future, quite a large number of authentications running through the PEPS are to be expected. This, however, can lead to scalability issues at the data center, especially when imaging that the population of an entire country is able to use this PEPS. Consequently, a move of the trusted PEPS service into the public cloud will considerably improve scalability. Nevertheless, a move of a trusted service into the public cloud clearly brings up new obstacles, especially with respect to citizens' privacy. Even when assuming that the PEPS in the public cloud works correctly, it still has to be assured that the cloud provider does neither learn nor leak any sensitive information. In the following, we propose a solution how the PEPS model can be securely realized in the public cloud by still preserving citizens' privacy. We thereby move both the S-PEPS and the C-PEPS into the public cloud.

4.1 Setup

For the setup of this cloud-based PEPS-PEPS scenario, similar to the current situation, we consider a trusted European Commission body to be responsible for managing and maintaining the PEPS metadata. In particular, the following metadata for accomplishing this cloud approach are required:

The European Commission body EC generates a UM-PRE and DSS key pair for every countries' S-PEPS and C-PEPS, resulting in $(\mathsf{rsk}_{\mathsf{S-PEPS_i}}, \mathsf{rpk}_{\mathsf{S-PEPS_i}})$ and $(\mathsf{sk}'_{\mathsf{S-PEPS_i}}, \mathsf{pk}'_{\mathsf{S-PEPS_i}})$, and $(\mathsf{rsk}_{\mathsf{C-PEPS_i}}, \mathsf{rpk}_{\mathsf{C-PEPS_i}})$ and $(\mathsf{sk}'_{\mathsf{C-PEPS_i}}, \mathsf{pk}'_{\mathsf{C-PEPS_i}})$ respectively. The EC, however, must ensure that the secret keys $\mathsf{rsk}_{\mathsf{S-PEPS_i}}$ and $\mathsf{rsk}_{\mathsf{C-PEPS_i}}$ are kept secret by the EC and cannot be accessed by $S-PEPS_i$ or $C-PEPS_i$.

Additionally, it provides every SP_j and IdP_k taking part in this cloud-based STORK architecture with appropriate UM-PRE keys: $(\mathsf{rsk}_{\mathsf{SP_j}}, \mathsf{rpk}_{\mathsf{SP_j}})$ and $(\mathsf{rsk}_{\mathsf{IdP_k}}, \mathsf{rpk}_{\mathsf{IdP_k}})$. Also the following re-encryption keys $\mathsf{rk}_{\mathsf{S-PEPS_i \to C-PEPS_i}}$, $\mathsf{rk}_{\mathsf{C-PEPS_i \to IdP_k}}$, $\mathsf{rk}_{\mathsf{C-PEPS_i \to S-PEPS_i}}$, $\mathsf{rk}_{\mathsf{S-PEPS_i \to SP_j}}$ are generated. The DSS key pairs ensuring integrity and authenticity of outgoing messages from either SP_j or IdP_k are created by the individual entities themselves, resulting in the key pairs $(\mathsf{sk}'_{\mathsf{SP_j}}, \mathsf{pk}'_{\mathsf{SP_j}})$ and $(\mathsf{sk}'_{\mathsf{IdP_k}}, \mathsf{pk}'_{\mathsf{IdP_k}})$, where the corresponding certificates must be publicly available.

To support the same logging and audit capability as in the current approach, EC additionally generates re-encryption keys for itself $\mathsf{rk}_{\mathsf{S-PEPS_i \to EC}}$ and $\mathsf{rk}_{\mathsf{C-PEPS_i \to EC}}$. During the authentication process, the individual $S-PEPS_i$ and $C-PEPS_i$ log the same data as in the current approach, but in encrypted form using either key $\mathsf{rk}_{\mathsf{S-PEPS_i \to EC}}$ or $\mathsf{rk}_{\mathsf{C-PEPS_i \to EC}}$. This allows the EC to inspect or audit transaction data if required.

4.2 Process Flow

In the following we discuss the detailed process flow when moving the individual PEPS instances into the public cloud. Again, we use a simplified notation for the individual entities (*SP* instead of SP_j, etc.) to describe the process flow of one particular authentication. The individual process steps are basically conform to the current PEPS-PEPS process flow. However, we generally assume that "attribute selection" (Steps 6 and 7 in Figure 2) and "giving consent" (Steps 11 and 12 in Figure 2) are carried out by the IdP for reasons that will be explained in Section 5.2.

1) Access protected resource: Same as in current situation.

2) SP Authentication Request: This step is similar to the current situation. However, instead of including the data `reqAttr`, `reqQAA`, and `sp_name` in plain in the SP authentication request, the SP computes $c_{S-PEPS} = \mathcal{E}_{\mathcal{R}}(\mathsf{rpk}_{\mathsf{S-PEPS}}, \mathtt{reqAttr} \| \mathtt{reqQAA} \| \mathtt{sp\_name})$ and signs this ciphertext resulting in $\sigma_{SP} = \mathcal{S}(\mathsf{sk}'_{\mathsf{SP}}, c_{S-PEPS})$. Both the signature σ_{SP} and the ciphertext c_{S-PEPS} are transmitted to the S-PEPS.

3) Select home country: Same as in current situation.

4) Provide home country information: Same as in current situation.

5) S-PEPS Authentication Request: The S-PEPS takes the ciphertext c_{S-PEPS} received from the SP, re-encrypts it for the C-PEPS using key $\mathsf{rk}_{\mathsf{S-PEPS \to C-PEPS}}$ resulting in c_{C-PEPS}, and finally signs it using its private key $\mathsf{sk}'_{\mathsf{S-PEPS}}$ resulting in $\sigma_{S-PEPS} = \mathcal{S}(\mathsf{sk}'_{\mathsf{S-PEPS}}, c_{C-PEPS})$. Depending on `country`, the S-PEPS forwards its authentication request including the data $(\sigma_{S-PEPS}, c_{C-PEPS})$ to the C-PEPS, which is responsible for authentications of `country`.

6) Show requested attributes: The C-PEPS verifies σ_{S-PEPS}. Additionally, the C-PEPS re-encrypts c_{C-PEPS} for the IdP using $\mathsf{rk}_{\mathsf{C-PEPS \to IdP}}$ and signs the resulting ciphertext c_{IdP} using key $\mathsf{sk}'_{\mathsf{C-PEPS}}$. Both results $(c_{IdP}, \sigma_{C-PEPS})$ are transmitted to the IdP. The IdP verifies σ_{C-PEPS}, decrypts c_{IdP}, and presents the citizen `reqAttr` and `sp_name`.

7) Confirm/deselect requested attributes: Similar to the current situation. Instead of the C-PEPS, the IdP carries out this process step.

8) Connect to appropriate IdP: This step is now included in step 6.

9) Authenticate: Same as in current situation.

10) Return identity and authentication data: This step is now included in step 12.

11) Get consent: Instead of the C-PEPS, the IdP sends `attr` to the citizen to get consent for further transmission to the C-PEPS and SP subsequently.

12) Give consent: Same as in current situation. Additionally, the IdP encrypts `attr` and `QAA` for the C-PEPS resulting in $c'_{C-PEPS} = \mathcal{E}_{\mathcal{R}}(\mathsf{rpk}_{\mathsf{C-PEPS}}, \mathtt{attr} \| \mathtt{QAA})$. Subsequently, the IdP generates the signature $\sigma_{IdP} = \mathcal{S}(\mathsf{sk}'_{\mathsf{IdP}}, c'_{C-PEPS})$ and transfers $(\sigma_{IdP}, c'_{C-PEPS})$ to the C-PEPS.

13) C-PEPS Authentication Response: This step is similar to the current situation. However, in this cloud-based approach, the C-PEPS verifies σ_{IdP}, re-encrypts c'_{C-PEPS} for the S-PEPS resulting in $c'_{S-PEPS} = \mathcal{E}_{\mathcal{R}}(\mathsf{rk}_{\mathsf{C-PEPS \to S-PEPS}}, c'_{C-PEPS})$, and signs c'_{S-PEPS} using key $\mathsf{sk}'_{\mathsf{C-PEPS}}$. The signature denoted as σ'_{C-PEPS} and the ciphertext c'_{S-PEPS} are transmitted to the S-PEPS.

14) S-PEPS authentication response: The S-PEPS verifies σ'_{C-PEPS} and re-encrypts c'_{S-PEPS} for the SP using key $\mathsf{rk}_{\mathsf{S-PEPS \to SP}}$. Finally, the S-PEPS signs the resulting ciphertext c_{SP} using key $\mathsf{sk}'_{\mathsf{S-PEPS}}$. $(\sigma'_{S-PEPS}, c_{SP})$ are transmitted to the SP.

15) Grant access: The SP verifies σ'_{S-PEPS} and decrypts c_{SP} to extract `attr` and `QAA`. The SP checks the `QAA` the citizen has been authenticated with and based on the attributes `attr` received, the SP either

grants or denies access to the requested service to the citizen.

Using PKE instead of UM-PRE: Although in the presented approach the key management using UM-PRE is less complex than using PKE, the use of UM-PRE may require stronger trust assumptions. Particularly, a strong trust must be put on the individual SP_i as in case of a fraudulent cooperation of one single SP_i with any $S-PEPS_i$ may break the UM-PRE infrastructure and allow the $S-PEPS_i$ to channel off or disclose any data ($\texttt{attr}\|\texttt{QAA}$ of the individual citizen) processed by or flowing through $S-PEPS_i$[5]. This issue is particular risky in a cloud environment, as the individual entities are easier to access via the Internet as in a trusted environment and consequently the attack surface is larger.

In order to reduce this risk, we propose an alternative based on conventional PKE. In this setting, the individual SP_i additional generate a one-time use PKE key pair $(\mathsf{sk}_{\mathsf{SP_i}}, \mathsf{pk}_{\mathsf{SP_i}})$. According to the proposed process flow, the public key $\mathsf{pk}_{\mathsf{SP_i}}$ is included in the encrypted SP authentication request sent to the S-PEPS in Step 2. The key is further forwarded to the C-PEPS and IdP in Step 5. After successful citizen authentication, in Step 12, the IdP encrypts attr and qaa for the SP resulting in $c'_{SP} = \mathcal{E}(\mathsf{pk}_{\mathsf{SP}}, \texttt{attr}\|\texttt{qaa})$. Additionally, c'_{C-PEPS} now includes the result of $\mathcal{E}_{\mathcal{R}}(\mathsf{rpk}_{\mathsf{C-PEPS}}, c'_{SP})$ instead of $\mathcal{E}_{\mathcal{R}}(\mathsf{rpk}_{\mathsf{C-PEPS}}, \texttt{attr}\|\texttt{QAA})$. For the subsequent Steps 13 and 14 also c'_{SP} instead of $\texttt{attr}\|\texttt{qaa}$ is used for re-encryption. In Step 14, the SP now takes $\mathsf{sk}_{\mathsf{SP}}$ to decrypt c'_{SP} and to extract $\texttt{attr}\|\texttt{QAA}$ to be used for the authentication decision in Step 15.

The advantage of this alternative is that in case of a fraudulent cooperation between SP_i and $S-PEPS_i$ only disclosure of c'_{SP} instead of $\texttt{attr}\|\texttt{QAA}$ is possible, which does not leak any sensitive information. In addition, by using a one-time use PKE key pair no further registration or complex setup is required.

5 EVALUATION

In this section we evaluate our proposed cloud-based PEPS-PEPS approach firstly with respect to privacy and security aspects, and, secondly, with respect to operational and practicability issues.

[5]We are aware that this risk also relates to the connection $C-PEPS_i$ and IdP_i. However, we consider an IdP_i acting trustworthier than an SP_i.

5.1 Security and Privacy Discussion

As already noted, we assume cloud providers are acting *honest but curious* and we are interested which personal and thus sensitive information, e.g., attr, are disclosed to an S-PEPS or C-PEPS operated in the public cloud. Table 1 compares the information seen by the individual entities (SP, S-PEPS, C-PEPS, IdP) in the current PEPS-PEPS scenario with the cloud-based scenario.

Note that the SP, in both approaches, clearly obtains all citizen information, since they are required for successfully providing its services.

The S-PEPS, acting as first gateway, also sees all citizen information in the current approach. Additionally, it gains knowledge on the country the citizen originates from. In contrast, the originating country remains the only information visible to the S-PEPS in the cloud-based approach. All other information is only available in encrypted form. With just the information of the country of origin, the S-PEPS is not able to determine the identity of any authenticating citizen.

For the current approach, also the C-PEPS sees all citizen information. Adopting our cloud-based approach, the C-PEPS is able to inspect processed data in encrypted form only, which does not allow for any personal data disclosure.

Finally, the IdP is considered to be trusted in the current and the cloud-based approach. The IdP gets knowledge on reqAttr, reqQAA, sp_name, attr, and QAA. Compared to the current approach, in the cloud-based approach the IdP gets to know which particular SP the information is provided to.

5.2 Practicability Discussion

In this section we discuss our proposed cloud approach based on selected criteria relating to practicability.

Re-Use of Existing Infrastructure: One criterion we focused on when having designed our cloud approach was to keep as much as of the existing infrastructure unaltered or unchanged. In general, main parts of the existing infrastructure can be kept untouched. This is particular important for the individual countries' eID infrastructures, as no new eID solution needs to be rolled out, which follows the general objective of STORK. The remaining entities (SP, S-PEPS, C-PEPS, IdP) need to be adapted to support UM-PRE. The STORK protocol does not necessarily need to be extended as SAML already supports the transfer of encrypted attributes and messages out of the box.

Conformance to Current Process Flow: When designing our cloud-based approach, we took care on

Table 1: Comparison of personal data disclosure between the current and the cloud-based PEPS-PEPS approach

Approach / Entitity	SP	S-PEPS	C-PEPS	IdP
Current	reqAttr, reqQAA, sp_name, attr, QAA	reqAttr, reqQAA, sp_name, attr, QAA, country	reqAttr, reqQAA, sp_name, attr, QAA	reqAttr, reqQAA, sp_name, attr, QAA
Cloud	reqAttr, reqQAA, sp_name, attr, QAA	country	×	reqAttr, reqQAA, sp_name, attr, QAA

staying conform with the current process flow. Basically, for supporting the cloud approach no severe changes in the communication and authentication process flow are necessary. However, Step 7 (*Confirm/deselect requested attributes*) and Steps 11 and 12 (*Get consent* and *Give consent)* must be carried out at the IdP when operating the C-PEPS in a public cloud. The reason is that otherwise the C-PEPS could present the citizen a list of different attributes than actually requested from the IdP. In this case, the C-PEPS could provide the SP with more citizen information than actually required.

Scalability: The main objective of our work was to design a scalable PEPS-PEPS scenario as increased and high usage of STORK can be expected in the future. This objective is mainly realized by moving important components, where high load can be expected, such as the S-PEPS and the C-PEPS, into the public cloud. Load bottlenecks at the SP or IdP are rather unlikely, because not all citizens are going to use the same SP or IdP at the same time.

Governance Structure: The governance structure of the STORK framework is currently in its detailed setup process. Thereby, a European Commission body will manage and maintain the individual PEPS metadata. This metadata includes the URLs and the individual signature certificates of the countries' PEPS. For supporting our cloud-based approach, this governance structure needs to be extended. In particular, the European Commission body additionally will be responsible for managing and issuing appropriate encryption keys based on a public key infrastructure (PKI) to the individual PEPS. However, the effort for these tasks can be kept within reasonable limits as we consider in the cloud approach to have one single S-PEPS and C-PEPS per country only.

6 CONCLUSIONS

The STORK framework – allowing secure citizen identification and authentication across borders – will play a major role across Europe in the future. Currently, STORK is in its start up phase, but continuously increasing usage can be expected. However, increased usage can easily lead to scalability issues, especially in the centralized PEPS approach, as theoretically the entire population of a country will be able to use and run authentications through a countries' PEPS. To overcome such scalability bottlenecks, in this paper we presented a solution by moving centralized country PEPS into a public cloud, which considerably improves scalability. We further evaluated this solution with respect to citizens' privacy preservation and practicability issues. We finally conclude that in terms of privacy, no identifying citizen information will be disclosed to a PEPS in the public cloud when applying the proposed approach. Additionally, existing national eID infrastructures can be kept untouched and no major changes to the current authentication process flow are required. Finally, our proposed solution perfectly fits into the STORK governance structure, which is currently being established.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for their valuable feedback. The second author has been partly supported by the European Commission through project FP7-FutureID, grant agreement number 318424.

REFERENCES

An, J. H. (2001). Authenticated Encryption in the Public-Key Setting: Security Notions and Analyses. *IACR Cryptology ePrint Archive*, 2001:79.

Chow, S. S. M., Weng, J., Yang, Y., and Deng, R. H. (2010). Efficient Unidirectional Proxy Re-Encryption. In *AFRICACRYPT*, pages 316–332.

Green, M. and Ateniese, G. (2007). Identity-Based Proxy Re-encryption. In *ACNS*, pages 288–306.

Leitold, H. and Zwattendorfer, B. (2010). STORK: Architecture, Implementation and Pilots. In *ISSE*, pages 131–142.

STORK (2011a). STORK D5.7.3 Functional Design for PEPS, MW models and interoperability.

STORK (2011b). STORK D5.8.3b Interface Specification.

Zwattendorfer, B., Sumelong, I., and Leitold, H. (2013). Middleware Architecture for Cross-Border Identification and Authentication. *JIAS*, 8(2):107–118.