Background

The InCommon Federation (InCommon) currently requires every service provider to register an encryption (public) key in its medata. This encryption key is used by an IdP to encrypt SAML assertions bound for the SP.

Until Baseline Expectations 2 (BE2), InCommon did not require an SP to encrypt its connection endpoints at the transport layer (TLS/SSL encryption, or https:// endpoints). Being able to encrypt SAML assertions at the SAML message level was critical to protect user information while in transit.

With InCommon officially transitioning to BE2, all InCommon SPs are now required to encrypt its endpoints at the transport layer. It brings into question whether message level encryption is still essential to all transactions.

Today's Gap - Community Assumption vs Reality

Even though InCommon requires an SP to register an encryption key, it does not explicitly require all SAML assertions to be encrypted at the message level. We as a community have assumed that:

a. The decision to encrypt at run time is at the IdP's discretion,
b. The SP should support message level encryption when the IdP does so, and that
c. By registering an encryption key, the SP has implemented the appropriate solution to process an encrypted message

As it turns out, service providers do not always properly support message level encryption even though they registered an encryption key. A lack of explicit requirement, compounded by the fact that there is not (yet) an obvious way for InCommon to directly verify every SP's implementation, means that we don't detect this disconnect until an IdP sends an encrypted transaction to the SP. That may not happen for quite some time since not every IdP insists on encrypting its transactions. This gap has led to confusion, disagreements, implementation delays, and poor user experience.

(exhibit: Cal Poly Pomona's recent dispute with Questica)

Questions to CTAB, TAC, and Community

1. With the BE2 transport layer endpoint encryption requirement in place, should InCommon continue to require all SPs to register a message level encryption key (therefore implying it supported message level encryption)?

Argument For	Argument Against
• TLS encryption is insufficient if the TLS processing occurs at a different network location than the resource, and that the data might be transmitted over unencrypted channels behind the TLS processor. • Some IdP may still want to require message level encryption. We should require all SPs to support it just in case. • more?	• TLS encryption is sufficient; message level encryption is unnecessary overhead for most use cases • Unless we are prepared to implement testing/validation requirements ($$$$$), we can't guarantee that an SP has deployed the correct implementation. That leads to hidden breakages down the line. The inconsistencies hurt the federation more than the benefits. • more?

Decisions/Actions

<tbd>

2. If we agree an SP must continue to supply an encryption key in metadata, what do we need to change/amend to close the implementation gap?

A. Clarify the SP's implementation requirements to support message level encryption

B. Devise testing mechanisms to validate implementation at registration time

C. Amend Baseline Expectations to include this clarification

D. Do nothing

E. Other

3. If we agree InCommon should no longer require an encryption key in SP metadata, what needs to happen?

3.1 How does an IdP who wants to require an SP to support message level encryption do so? Or is that no longer a "InCommon" concern?

3.2 Should InCommon still clarify/enforce implementation requirements for an SP "if" it indicated support for message level encryption?

(This then falls into the "if you are going to do it, do it this way" category.)

3.3 Are there any inter-federation implications? e.g., compatibility with other entities in eduGAIN

3.4 When introducing this type of potential change to federation operating and integration policy/practices/requirements, what is the decision making process and the RACI?