Groupe de travail Réseau
Request for Comments : 4401
Catégorie : Sur la voie de la normalisation

N. Williams, Sun Microsystems
février 2006
Traduction Claude Brière de L'Isle

Extension d'API de fonction pseudo aléatoire pour l'interface de programme d'application de service de sécurité générique (GSS-API)

Statut de ce mémoire
Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document définit une extension de fonction pseudo aléatoire (PRF, *Pseudo-Random Function*) à l'interface d'application de service de sécurité générique (GSS-API, *Generic Security Service Application Program Interface*) pour les protocoles d'application de chiffrement dans un contexte de sécurité de GSS-API établi donné. La principale utilisation prévue de cette fonction est de sécuriser les couches de session qui n'utilisent pas ou ne peuvent pas utiliser la vérification d'intégrité de message (MIC, *message integrity check*) GSS-API par message et enveloppent des jetons pour la protection de la session.

Table des matières

1. Introduction

Il est apparu un besoin pour les utilisateurs de GSS-API de chiffrer les protocoles cryptographiques des applications en utilisant les contextes établis de sécurité de GSS-API. De telles applications peuvent utiliser la GSS-API [RFC2743] pour l'authentification, mais pas pour la sécurité du transport (qu'elles qu'en soient les raisons) et comme GSS-API ne fournit pas de méthode pour obtenir de matériel de chiffrement des contextes de sécurité établis, ces applications ne peuvent pas faire une utilisation effective de GSS-API.

Pour répondre à ce besoin, on définit une extension de fonction pseudo aléatoire (PRF, *pseudo-random function*) à la GSS-API.

Bien que le présent document spécifie une API abstraite comme extension à la GSS-API version 2, mise à jour 1, et bien qu'il spécifie les liens de cette extension au langage de programmation C, il ne spécifie pas une révision de GSS-API et donc ne traite pas la question de savoir comment les applications portables détectent la prise en charge et s'assurent de l'accès à cette extension. On renvoie cette question à une future mise à jour complète de GSS-API.

1.1 Conventions utilisées dans ce document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. GSS_Pseudo_random()

Entrées :
- o bride de contexte CONTEXT,
- o ENTIER prf_key,
- o CHAINE D'OCTETS prf_in,
- o ENTIER desired_output_len *(longueur de résultat désirée)*

Résultats :
- o ENTIER major_status,
- o ENTIER minor_status,
- o CHAINE D'OCTETS prf_out

Codes de retour major_status :
- o GSS_S_COMPLETE indique qu'il n'y a pas d'erreur.
- o GSS_S_NO_CONTEXT indique qu'un contexte nul a été fourni en entrée.
- o GSS_S_CONTEXT_EXPIRED indique qu'un contexte expiré a été fourni en entrée.
- o GSS_S_UNAVAILABLE indique que le mécanisme ne prend pas en charge cette fonction ou, si le contexte de sécurité n'est pas complètement établi, que le contexte n'est pas prêt à calculer la PRF avec la prf_key donnée, ou que la prf_key donnée n'est pas disponible.
- o GSS_S_FAILURE indique une défaillance générale, éventuellement due aux données d'entrée qui sont trop grandes ou de longueur zéro, ou parce que la longueur du résultat désiré est zéro ; le code de statut mineur peut fournir des informations supplémentaires.

Cette fonction applique la fonction pseudo aléatoire (PRF, *pseudo-random function*) chiffrée du mécanisme du contexte établi aux données d'entrée ('prf_in') chiffrées avec le matériel de clé associé au contexte de sécurité donné et identifié par "prf_key", et donne en sortie la chaîne d'octets résultante ('prf_out') de la longueur de sortie désirée.

La longueur minimum des données d'entrée est d'un octet.

Les mécanismes DOIVENT être capables de consommer toutes les données d'entrées fournies dans prf_in qui font 2^14 octets ou moins.

Si un mécanisme ne peut pas consommer autant de données d'entrée que fournies par l'appelant, alors GSS_Pseudo_random() DOIT retourner GSS_S_FAILURE.

La longueur minimum de "desired_output_len" est un.

Les mécanismes DOIVENT être capables de sortir au moins jusqu'à 2^14 octets.

Si la mise en œuvre ne peut pas produire le résultat désiré à cause d'un manque de ressources, elle DOIT alors retourner GSS_S_FAILURE et DOIT établir un code d'état mineur convenable.

La prf_key peut prendre les valeurs suivantes : GSS_C_PRF_KEY_FULL, GSS_C_PRF_KEY_PARTIAL, ou des valeurs spécifiques du mécanisme, si il en est. Ce paramètre est destiné à distinguer les meilleures clés cryptographiques qui peuvent n'être disponibles qu'après l'établissement complet du contexte de sécurité et les clés qui peuvent être disponibles avant le plein établissement du contexte de sécurité. Pour certains mécanismes, ou contextes, ces deux valeurs de prf_key PEUVENT se référer aux mêmes clés de chiffrement ; pour des mécanismes comme la GSS-API Kerberos V [RFC1964] où un homologue peut affirmer une clé qui peut être considérée comme meilleure que les autres, elles PEUVENT être des clés différentes.

GSS_C_PRF_KEY_PARTIAL correspond à une clé qui aurait été utilisée alors que le contexte de sécurité était partiellement établi, même si il est pleinement établi quand GSS_Pseudo_random() est effectivement invoqué. Les valeurs

de prf_key spécifiques du mécanisme sont destinées à se référer à toutes les autres clés qui peuvent être disponibles.

La valeur GSS_C_PRF_KEY_FULL correspond à la meilleure clé disponible pour les contextes de sécurité pleinement établis.

GSS_Pseudo_random() a les propriétés suivantes :

- sa chaîne de sortie DOIT être une fonction pseudo aléatoire [GGM1], [GGM2] de l'entrée chiffrée avec le matériel de clé provenant du contexte de sécurité donné -- les chances d'obtenir le même résultat avec des paramètres d'entrée différents devraient être exponentiellement petites.

- quand elle est appliquée avec succès aux mêmes entrées par un initiateur et un acceptant qui utilisent le même contexte de sécurité, elle DOIT produire les _mêmes résultats_ pour l'initiateur et l'acceptant, même si elle est invoquée plusieurs fois (tant que le contexte de sécurité n'est pas expiré).

- à l'établissement complet d'un contexte de sécurité, toutes les clés de chiffrement et/ou négociations utilisées pour calculer la PRF avec toute prf_key DOIVENT être authentifiées (mutuellement, si l'authentification mutuelle est mise en œuvre pour ce contexte de sécurité).

- les résultats du GSS_Pseudo_random() du mécanisme (pour les différentes entrées) et ses jetons par message pour le contexte de sécurité donné DOIVENT être "séparés cryptographiquement" ; en d'autre termes, il ne doit pas être possible de récupérer le matériel de clé pour une opération de mécanisme ou de transformer ses jetons et résultats de PRF d'un à l'autre de ces jetons et résultats de PRF. (C'est une autre façon de dire que la déduction de clé et des opérations et constructions cryptographiques fortes doivent être utilisées.)

- comme impliqué par les exigences ci-dessus, il NE DOIT PAS être possible d'accéder à des clés brutes d'un contexte de sécurité à travers GSS_Pseudo_random(), quelles que soient les entrées.

2.1 Liens C

```
#define GSS_C_PRF_KEY_FULL 0
#define GSS_C_PRF_KEY_PARTIAL 1

OM_uint32 gss_pseudo_random(
  OM_uint32                *minor_status,
  gss_ctx_id_t             context,
  int                      prf_key,
  const gss_buffer_t       prf_in,
  ssize_t                  desired_output_len,
  gss_buffer_t             prf_out
);
```

Codes d'état majeur suplémentaires pour les liens C :

- GSS_S_CALL_INACCESSIBLE_READ

- GSS_S_CALL_INACCESSIBLE_WRITE

Voir la [RFC2744].

3. Considérations relatives à l'IANA

Ce document n'a pas actuellement de considérations relatives à l'IANA. Si et quand sera créé un registre pertinent de l'IANA pour les symboles GSS-API, les noms de fonctions génériques et spécifiques du langage, les noms de constantes, et les valeurs de constantes décrites ci-dessus devraient être ajoutées à un tel registre.

4. Considérations sur la sécurité

Il faut faire attention à concevoir de façon appropriée la fonction PRF d'un mécanisme.

Les fonctions de PRF d'un mécanisme GSS devraient utiliser une clé déduite des clés de session authentifiées d'un contexte et devraient préserver les propriétés de sécurité vers l'avant des échanges de clé du mécanisme.

Certains mécanismes peuvent prendre en charge la fonction PRF de GSS avec des contextes de sécurité qui ne sont pas pleinement établis, mais les applications DOIVENT supposer que l'authentification, mutuelle ou autre, ne s'est pas achevée tant que le contexte de sécurité n'est pas pleinement établi.

Les appelants de GSS_Pseudo_random() devraient éviter de l'invoquer accidentellement avec les mêmes entrées. Une technique utile est d'ajouter devant la chaîne prf_in, par convention, une chaîne indiquant l'objet auquel est destiné le résultat de la PRF d'une façon telle que des contextes uniques dans lesquels la fonction est invoquée lui donnent des entrées uniques.

Les fonctions pseudo aléatoires sont, par nature, capables de produire seulement des quantités limitées de résultats cryptographiquement sûrs. La quantité exacte de résultat qu'on peut utiliser en toute sécurité varie malheureusement d'une PRF à l'autre (ce qui empêche de recommander des nombres spécifiques). À cause de cela, on recommande que sauf si on sait réellement ce qu'on fait (c'est-à-dire, si on est un cryptographe qualifié pour avoir un jugement sur les fonctions cryptographiques dans les domaines de la période, présence de cycles courts, etc.) on limite la quantité de résultat de PRF utilisé au minimum nécessaire. Voir dans la [RFC4086] plus d'informations sur les "Exigences d'aléa pour la sécurité".

Pour certains mécanismes, le coût de calcul de GSS_Pseudo_random() peut augmenter significativement lorsque la longueur des données de prf_in et/ou la longueur de résultat désiré augmente. Cela signifie que si une application peut être conduite à fournir de très grosses chaînes d'octets d'entrée et à demander de très longues chaîne d'octets de résultat, cela peut constituer une attaque de déni de service sur l'application ; donc, les applications DEVRAIENT fixer des limites appropriées à la taille de toute chaîne d'octets d'entrées reçue de ses homologues sans protection de l'intégrité.

5. Références

5.1 Références normatives

[GGM1] Goldreich, O., Goldwasser, S., and S. Micali, "How to Construct Random Functions", Journal of the ACM, octobre 1986.

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2743] J. Linn, "Interface générique de programme d'application de service de sécurité, version 2, mise à jour 1", janvier 2000. (*MàJ par* RFC5554)

[RFC2744] J. Wray, "API de service générique de sécurité, version 2 : liaisons C", janvier 2000. *(P.S.)*

5.2 Références pour information

[GGM2] Goldreich, O., Goldwasser, S., and S. Micali, "On the Cryptographic Applications of Random Functions", Proceedings of CRYPTO 84 on Advances in cryptology, 1985.

[RFC1964] J. Linn, "Mécanisme GSS-API de Kerberos version 5", juin 1996. (*MàJ par* RFC4121 et RFC6649)

[RFC4086] D. Eastlake 3rd, J. Schiller, S. Crocker, "Exigences d'aléa pour la sécurité", juin 2005. (*Remplace* RFC1750) (BCP0106)

Adresse de l'auteur

Nicolas Williams
Sun Microsystems
5300 Riata Trace Ct
Austin, TX 78727
US

mél : Nicolas.Williams@sun.com

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est actuellement fourni par la Internet Society.