Implémenter une authentification sécurisée sans être un expert en cryptographie

par Christophe Tremlet, Directeur Exécutif de Division, Micros & Sécurité, Maxim Integrated

Aujourd'hui, la sécurité numérique est l'un des sujets les plus en vue de la conception électronique. Pour de nombreux ingénieurs, "chiffrement" (ou cryptage) est vraisemblablement le premier mot qui vient à l'esprit quand ils pensent sécurité. Il est peu probable qu'ils pensent en priorité à l'authentification.

Pourtant, l'authentification est une composante essentielle des dispositifs et des transactions sécurisés. Prenons l'exemple des opérations bancaires en ligne. Clairement, vous souhaitez que les informations confidentielles telles que vos soldes et vos numéros de compte, soient chiffrées. C'est ce qui se produit quand votre navigateur internet affiche le cadenas vert à côté de https://. Ceci dit, la première chose que vérifie le navigateur internet lors de l'établissement d'une connexion sécurisée est que le site web de la banque est bien authentique ; en d'autres termes, il *authentifie* le site web de la banque. En effet, envoyer son login et son mot-de-passe à un faux site serait extrêmement nuisible, puisque ces informations de connexion pourraient être réutilisées pour effectuer toutes sortes de transactions illicites au nom du détenteur du compte qui ne se douterait de rien. La navigation internet sécurisée est généralement assurée par le protocole TLS/SSL, qui garantit l'authenticité et la confidentialité.

L'authentification est également importante pour les applications IoT (Internet of Things, ou internet des objets) : un nœud qui n'est pas digne de confiance peut mettre en danger toute une infrastructure. Envisageons par exemple les compteurs électriques communiquants. Un moyen simple pour un attaquant de s'en prendre au réseau, est d'injecter un virus ou un malware dans les compteurs intelligents. Les compteurs infectés peuvent alors envoyer de faux messages à l'infrastructure reflétant une consommation très différente de la réalité. Le réseau de distribution peut alors se trouver déséquilibré ; au pire, l'attaque peut provoquer une coupure électrique de grande envergure. Afin d'éviter cette situation, le matériel, comme le firmware (micrologiciel) du compteur, doivent être vérifiés comme étant authentiques. Le processus d'authentification du micrologiciel s'appelle *secure boot (amorçage sécurisé)*.

Implémentation d'une méthode d'authentification efficace

Maintenant que nous avons compris l'importance de l'authentification, voyons comment la mettre en œuvre. La manière la plus triviale d'authentifier est d'utiliser un mot de passe. Dans l'exemple de notre compteur intelligent, le dispositif pourrait envoyer un mot de passe au système de commande du réseau de distribution. Le serveur vérifierait alors le mot-de-passe, puis autoriserait les transactions. Si cette méthode est facile à comprendre, elle est loin d'être la meilleure. Un attaquant peut très facilement espionner la communication, enregistrer le mot de passe, et le réutiliser pour authentifier un équipement non-authentique. Pour cette raison, nous considérons l'authentification par mot-de-passe comme faible.

Une bien meilleure façon de réaliser l'authentification dans le monde numérique, est la méthode dite *"challenge-response" (ou défi-réponse)*. Jetons un oeil à deux différents types de défi-réponse : l'un basé sur la cryptographie symétrique et l'autre basé sur la cryptographie asymétrique.

L'authentification à cryptographie symétrique s'appuie sur un secret partagé. L'hôte et le dispositif à authentifier détiennent un même nombre secret. L'hôte envoie d'abord un nombre aléatoire, le *défi*, au dispositif. Le dispositif calcule une *signature numérique* fonction du secret et du défi, et le renvoie à l'hôte.
L'hôte exécute lui aussi le calcul et compare les résultats. Si les deux calculs fournissent le même résultat, le dispositif est authentifié (Figure 1). Afin de s'assurer que le résultat ne puisse être imité, il est indispensable d'utiliser une fonction disposant des propriétés mathématiques adéquates ; par exemple, il faut absolument veiller à ce qu'il soit impossible de récupérer le secret à partir du résultat du calcul. Les fonctions hash sécurisées, comme SHA-256, répondent à ce besoin. Dans le cas de la méthode défi-réponse, le dispositif prouve qu'il connait bien le secret sans avoir à le révéler. Même si un attaquant venait à intercepter la communication, il ne pourrait pas accéder au secret partagé.

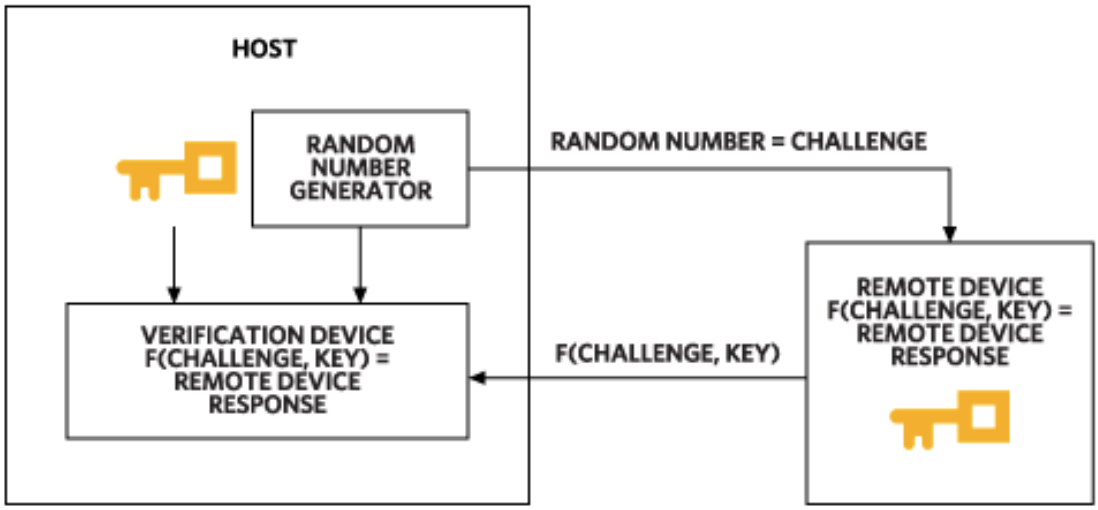


Figure 1 : L'authentification à cryptographie symétrique s'appuie sur un nombre secret partagé entre l'hôte et le dispositif.

L'authentification à cryptographie asymétrique s'appuie sur deux clés : une clé privée et une clé publique. La clé privée n'est connue que du dispositif à authentifier, tandis que la clé publique peut être révélée à toute entité désirant authentifier le dispositif. Comme dans le cas de la méthode évoquée auparavant, l'hôte envoie un défi au dispositif. Le dispositif calcule une signature fonction du défi et de la clé privée, et la renvoie à l'hôte (Figure 2). Mais ici, l'hôte utilise la clé publique pour vérifier la signature. Il est également indispensable que la fonction qui sert à calculer la signature dispose de certaines propriétés mathématiques. Les fonctions les plus couramment utilisées pour les systèmes à

cryptographie asymétrique sont RSA et ECDSA. Là encore, le dispositif démontre qu'il connaît le secret, en l'occurrence la clé privée, sans la révéler.

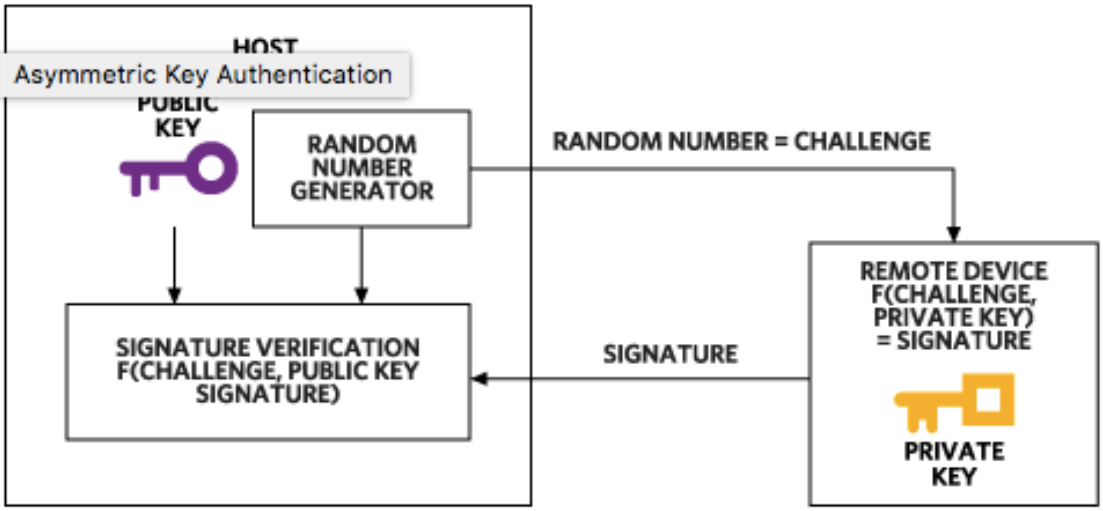


Figure 2 : L'authentification à clé asymétrique s'appuie sur une clé publique et une clé privée.

Qu'apportent les circuits intégrés en terme d'authentification ?

L'authentification défi-réponse nécessite que l'objet à authentifier contienne un secret. Dans le cas de la cryptographie symétrique, il s'agit du secret partagé entre l'hôte et le dispositif. Dans le cas de la cryptographie asymétrique, il s'agit de la clé privée. Dans tous les cas, la sécurité apportée par l'authentification défi-réponse est compromise si le secret est révélé. C'est là que les CI de sécurité apportent leur valeur ajoutée. Une caractéristique fondamentale des CI de sécurité est d'apporter un niveau de protection élevé des clés et des secrets.

Maxim propose trois familles de solutions pour l'authentification :

- CI d'authentification : Il s'agit de dispositifs configurables à fonction fixe, qui offrent le moyen le plus économique de mettre en œuvre une authentification défi-réponse, avec un jeu limité d'opérations cryptographiques
- Microcontrôleurs sécurisés : En plus de permettre l'authentification défi-réponse, ces dispositifs offrent un jeu complet de fonctions de cryptographiques, notamment le chiffrement
- Microcontrôleurs basse-consommation : Ces produits ne ciblent pas exclusivement la sécurité, mais disposent de toutes les briques permettant la mise en œuvre d'une authentification forte

Parmi les CI d'authentification, les produits basés sur SHA-256 supportent l'authentification basée sur des secrets partagés (Figure 3), alors que ceux basés sur ECDSA utilisent une paire clé privée/clé publique (Figure 4). En plus de moteurs cryptographiques, ces produits disposent de mémoire EEPROM embarquée. Cette mémoire est configurable et peut servir à

stocker des données d'utilisateur authentifiées comme des paramètres d'étalonnage de capteurs.

Les produits basés sur SHA-256 sont les solutions les plus abordables. Leur avantage est de permettre l'authentification mutuelle. Par contre, la distribution du secret partagé nécessite certaines précautions pour que le secret ne se trouve pas exposé pendant la fabrication et l'installation du dispositif. Le secret peut être programmé dans une usine Maxim pour éviter cet inconvénient.

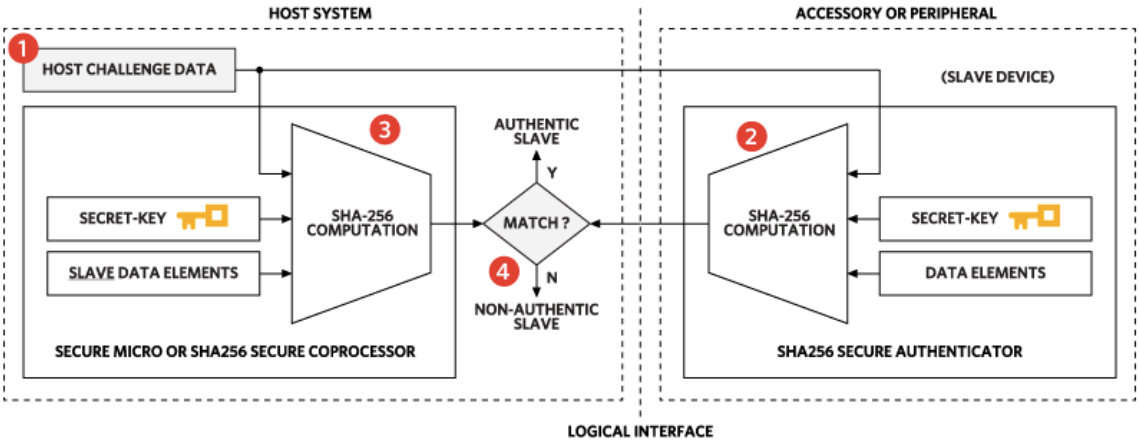


Figure 3 : L'authentification sécurisée SHA-256 est basée sur des secrets partagés.

Les CI Maxim DS28E15/22/25 sont basés sur la technologie SHA-256, et diffèrent par leur taille mémoire interne. Etant donné que le même secret est stocké à la fois côté hôte et côté dispositif, nous recommandons d'utiliser un coprocesseur comme le DS2465 côté hôte.

Les produits à cryptographie asymétrique, tels que les DS28C36 et DS28E35, proposent un implémentation plus souple, puisque la clé n'a pas besoin d'être protégée contre la divulgation côté hôte. Cependant, pour décharger le microcontrôleur côté hôte des calculs sur la clé publique et assurer d'autres opérations sécurisées, des coprocesseurs côté hôte, tels que le DS2476 (CI compagnon du DS28C36), sont disponibles pour simplifier le développement de la solution système.

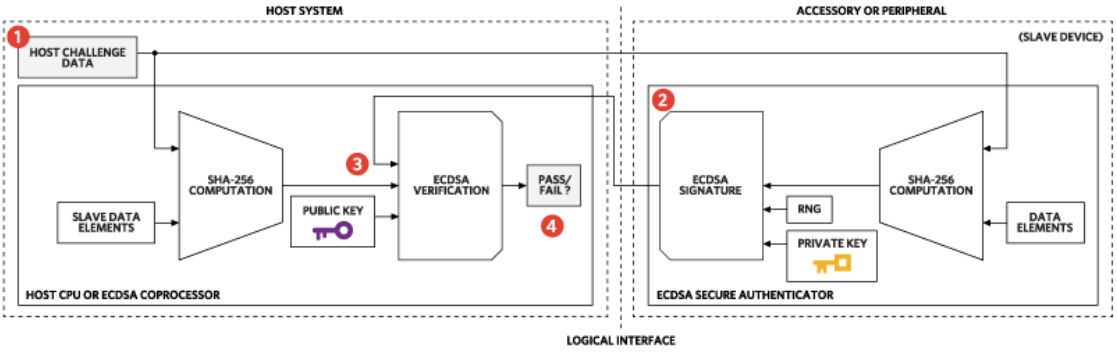


Figure 4 : Figure 4 : l'authentification à base d' ECDSA s'appuie sur une paire clé privée/clé publique.

Microcontrôleurs sécurisés avec support cryptographique symétrique et asymétrique
Maxim propose une large gamme de microcontrôleurs sécurisés allant du processeur applicatif MAX32590 (ARM9 à 384 MHz) capable d'exécuter un système d'exploitation évolué comme Linux, jusqu'aux coprocesseurs du type MAX32555 ou MAXQ1061.

Ces microcontrôleurs supportent la cryptographie symétrique et asymétrique, qui permettent la signature numérique et l'authentification, ainsi que les algorithmes de chiffrement. Ils disposent d'accélérateurs matériels pour SHA, RSA, ECDSA et AES, ainsi que d'une bibliothèque complète de cryptographie fournissant un API (Application Programming Interface, ou interface de programmation d'application) "clés en main", conforme aux différentes normes. Ils sont dotés d'un "secure boot" (amorçage sécurisé), de sorte que l'authenticité du micrologiciel est toujours garantie. Grâce à leur jeu complet de fonctions crypto, ils peuvent gérer différents schémas d'authentification.

Le MAXQ1061 est un coprocesseur qui non seulement permet l'authentification, mais gère aussi les étapes les plus critiques du protocole de communication sécurisé standard sur IP TLS/SSL. Gérer le protocole TLS au sein de la puce elle-même améliore le niveau de sécurité et décharge le processeur central de tâches de calcul intensif. Ceci s'avère précieux dans le cas de systèmes embarqués dotés de ressources limitées.

Les microcontrôleurs basse-consommation comme le MAX32626 ciblent les dispositifs wearables, et ne sont pas centrés sur la sécurité. Néanmoins, pour répondre aux attaques qui se font de plus en plus fréquentes, ce produit a été conçu en pensant aux défis que posera la sécurité de demain. Par conséquent, le MAX32626 dispose d'une TPU (Trust Protection Unit, ou unité de protection de confiance) supportant l'authentification, ainsi que l'AES matériel pour le chiffrement et l'amorçage sécurisé.

Résumé
Grâce à cet article, nous avons découvert ce qu'est l'authentification, et pourquoi cette fonction est aussi importante. Nous avons également pu voir que grâce aux solutions silicium existantes, on pouvait mettre en œuvre l'authentification sans être un expert de la cryptographie.