Practical Secure Aggregation for Privacy-Preserving Machine Learning

3.3, 3.5 and 3.6

Kun Woong Kim

Seoul National University

2020.7.29.

Contents

Introduction

To protect user's private information from adversaries, we need **encryption** algorithms.

The secure protocol of this paper contains two encryption algorithms; **Authenticated Encryption** and **Signature Scheme**.

Key concepts

Encryption is the process of encoding information or sensitive data so only authorized parties can access it.

Here, we take a look at the definition and configuration of them.

In fact, **the standard algorithms** which are guaranteed internationally are used to construct this protocol.

Notations

k : the security parameter of the scheme (a key size in bits)

m : a message (plaintext)

M : an adversary

c : a key $\in \{0,1\}^k$

d^{PK}, d^{SK} : public key and secret key

σ : a signature

3.3 Authenticated Encryption

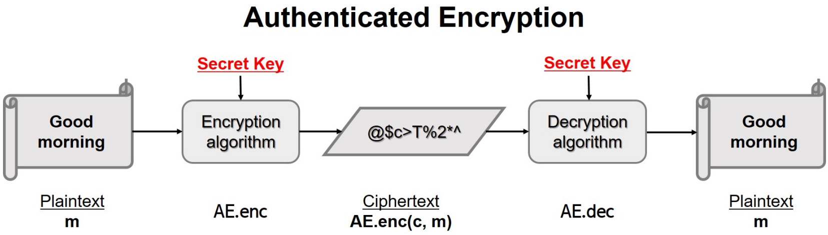
AE is a symmetric encryption that satisfies
confidentiality (by IND-CPA) and integrity (by INT-CTXT).

Symmetric encryption is a triple of three algorithms.

$$\textbf{AE} = (\mathcal{K}, \textbf{AE.enc}(), \textbf{AE.dec}())$$

1. A key generation algorithm $\mathcal{K}$ returns a secret key c.
2. An encryption algorithm **AE.enc** returns a ciphertext e.
   $\textbf{AE.enc}(c, m) \to e$
3. An decryption algorithm **AE.dec**
   $\textbf{AE.dec}(c, e) \to m \in \mathcal{M} \cup \{\perp\}$ (an error symbol)
   where $\mathcal{M}$ is a message space.

3.3 Authenticated Encryption



3.3 Authenticated Encryption

(1) Correctness

$$\textbf{AE.dec}(c, \textbf{AE.enc}(c, m)) = m$$

$\forall c \in \{0,1\}^k$ and $\forall m$.

3.3 Authenticated Encryption

(2) Security
"Authenticated" Encryption requires two security properties :
Confidentiality and **Integrity**

3.3 Authenticated Encryption

confidentiality

IND-CPA (Indistinguishability under Chosen Plaintext Attack)

Consider a probablistic game (Challenger vs. Adversary)
(i) generate $c \leftarrow \mathcal{K}$ (challenger)
(ii) M chooses and sends m_0 and m_1 to the challenger.
(iii) $e \leftarrow$ **AE.enc**(c, m_b), $b = 0$ or 1 (randomly chosen by the challenger)
(iv) M can query on chosen plaintext m_i to the challenger, and obtains $e_i =$ **AE.enc**(c, m_i)
(v) M guesses b' to be as $b' = b$, for $b \in \{0, 1\}$

For any PPT adversary M,

$$Pr(M \text{ wins}) = Adv_{\textbf{AE},M}(k) \leq \frac{1}{2} + \epsilon(k)$$

3.3 Authenticated Encryption

integrity

INT-CTXT (Indistinguishability under Ciphertext Integrity)

Consider a probablistic game (Challenger vs. Adversary)
(i) generate $c \leftarrow \mathcal{K}$
(ii) M can query on chosen messages m_i to the challenger and obtain $e_i \leftarrow \textbf{AE.enc}(c, m_i)$
(iii) If $\textbf{AE.dec}(c, e') \rightarrow m'$ for new m' and e',
Here, "new" means m' and e' have never been queried at (ii).
If $m' \neq \perp$, then M wins.

For any PPT adversary M,

$$Pr(M \text{ wins}) = Adv_{\textbf{AE},M}(k) \leq \epsilon(k)$$

for some negligable function $\epsilon(\cdot)$.

3.3 Authenticated Encryption

Advanced Encryption Standard (AES) is the standard encryption algorithm established by U.S. NIST in 2001.

It is included in ISO/IEC standard.

This paper used **AES-GCM** (Authenticated Encryption Standard Galois/Counter Mode) with 128-bit keys;

All user-to-user messages are authenticated through this AE scheme.

3.5 Signature Scheme

Signature Scheme is a mathematical technique to validate the authenticity and integrity of a message. It provides the assurances of **evidence of origin** and identity.

It is based on the asymmetric encryption algorithms.

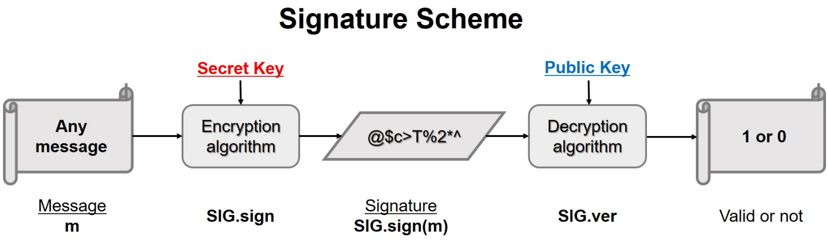
3.5 Signature Scheme

Signature Scheme is a triple of three algorithms.

$$\textbf{SIG} = (\textbf{SIG.gen}, \textbf{SIG.sign}, \textbf{SIG.ver})$$

1. A generation algorithm **SIG.gen** generates a secret key d^{SK} and a public key d^{PK}. (Here, k is the length of these keys.)
   $\textbf{SIG.gen}(k) \to (d^{PK}, d^{SK})$
2. A signing algorithm **SIG.sign** outputs a signature.
   $\textbf{SIG.sign}(d^{SK}, m) \to \sigma$
3. A verification algorithm $\textbf{SIG.ver}(d^{PK}, m, \sigma)$ returns 0 or 1.
   $\textbf{SIG.ver}(d^{PK}, m, \sigma) \to \begin{cases} 0 \text{ if } \sigma \text{ is invalid signature for } m. \\ 1 \text{ if } \sigma \text{ is valid signature for } m. \end{cases}$

3.5 Signature Scheme



3.5 Signature Scheme

(1) Correctness

$$\mathsf{Pr}(\textbf{SIG.ver}(d^{PK}, m, \sigma) = 1) = 1, \forall m$$

where $(d^{PK}, d^{SK}) \leftarrow \textbf{SIG.gen}(k)$ and $\sigma \leftarrow \textbf{SIG.sign}(d^{SK}, m)$

3.5 Signature Scheme

(2) Security :

UF-CMA (Unforgetability under chosen-message attack)

Consider a probablistic game (Challenger vs. Adversary)
(i) $(d^{PK}, d^{SK}) \leftarrow$ **SIG.gen(k)**
(ii) M can query on chosen messages m_i and obtain
$\sigma_i \leftarrow \textbf{SIG.sign}(d^{SK}, m_i)$
(iii) M wins iff $\textbf{SIG.ver}(d^{PK}, m', \sigma') = 1$ for new m' and σ'
Here, "new" means m' and σ' have never been queried at (ii).

For any PPT adversary M,

$$Pr(M \text{ wins}) = Adv_{\textbf{SIG},M}(k) \leq \epsilon(k)$$

for some negligable function $\epsilon(\cdot)$.

3.6 Public-Key Infrastructure

A public key infrastructure (PKI) is a set of roles, policies, hardware, software and procedures needed to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption.

We need public keys to construct a signature scheme, hence PKI is required also.