Mermory-efficient Fujisaki-Okamoto transformation

Ganyu (Bruce) Xu (g66xu)

May 8, 2024

1 Introduction

2 Preliminaries

2.1 Security definitions

2.2 Fujisaki-Okamoto transformation

3 Memory-efficient OW-PCVA transformation using MAC

In this section we propose an alternative OW-PCVA transformation T.

Let $\text{PKE} = (\text{KeyGen}, \text{Enc}, \text{Dec})$ be a public-key encryption scheme defined over key space $\mathcal{K}_{\text{PKE}}$, message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, and coin space $\mathcal{R}$. Let $\text{MAC} = (\text{Sign}, \text{Verify})$ be a message authentication code defined over key space $\mathcal{K}_{\text{MAC}}$, message space $\mathcal{M}$ (the same message space as PKE), and tag space $\mathcal{T}$. Let $G : \mathcal{M} \to \mathcal{R} \times \mathcal{K}_{\text{MAC}}$ be a random oracle. The transformed public-key encryption scheme is denoted by $\text{PKE}_1 = T(\text{PKE}, \text{MAC}, G)$ and includes the sub-routines $(\text{KeyGen}, \text{Enc}_1, \text{Dec}_1)$.

The transformed encryption and decryption routines are as follows

Algorithm 1: OW-PCVA encryption E_1

```
  Input: pk, $m \in \mathcal{M}$
1 $(r, k) \leftarrow G(m)$;
2 $c \leftarrow E(\text{pk}, m, r)$;
3 $t \leftarrow \text{MAC}(k, c)$ // "encrypt-then-mac";
4 return $(c, t)$;
```

Algorithm 2: OW-PCVA decryption D_1

```
  Input: $\text{sk}, c \in \mathcal{C}, t \in \mathcal{T}$
1 $\hat{m} \leftarrow D(\text{sk}, c)$;
2 $(\hat{r}, \hat{k}) \leftarrow G(\hat{m})$;
3 if $\text{MAC}(\hat{k}, c) \neq t$ then
4     return $\perp$;
5 end
6 return $\hat{m}$;
```

If PKE is OW-CPA secure and MAC is existentially unforgeable under chosen message attack, then the transformed encryption scheme PKE_q is OW-PCVA:

Theorem 3.1. *Let* PKE *be δ-correct and have γ-spread. For every OW-PCVA adversary $\mathcal{A}_1$ against* PKE_1 *that makes q_V ciphertext validation queries and q_G hash queries, and that has advantage ϵ_1, there exists an OW-CPA adversary against* PKE *with advantage ϵ_{PKE} and an EF-CMA adversary against* MAC *with advantage ϵ_{MAC} such that*

$$\epsilon_{\text{PKE}_1} \leq q_G \cdot \delta + q_V \cdot \epsilon_{\text{MAC}} + q_G \cdot \epsilon_{\text{PKE}}$$

The proof follows a sequence of games that begins with the standard OW-PCVA game but incrementally replaces the PCO and the CVO with alternative implementations that can be simulated by an OW-CPA adversary and/or the EF-CMA adversary who does not have the secret key.

Proof. Let $\mathcal{A}$ denote an adversary against the OW-PCVA security of Enc_1. Consider the following sequence of games.

Game G_0 is the standard OW-PCVA game. In the original OW-PCVA game, queries to the random oracle G are stored on a tape $\mathcal{L}^G$. By convention, we say that $G(m) = (r, k)$ if and only if $(m, r, k) \in \mathcal{L}^G$.

We can be more explicit with the implementation of the PCO and the CVO:

Algorithm 3: Standard PCO in G_0

```
Input: $m \in \mathcal{M}, (c \in \mathcal{C}, t \in \mathcal{T})$
1 $\hat{m} \leftarrow \text{Dec}(\text{sk}, c)$;
2 $(\hat{r}, \hat{k}) = G(\hat{m})$;
3 return $[\![m = \hat{m}]\!]$ and $[\![\text{MAC}(\hat{k}, c) = t]\!]$
```

Algorithm 4: Standard CVO in G_0

```
Input: $c \in \mathcal{C}, t \in \mathcal{T}$
1 $\hat{m} \leftarrow \text{Dec}(\text{sk}, c)$;
2 $(\hat{r}, \hat{k}) \leftarrow G(\hat{m})$;
3 return $[\![\text{MAC}(\hat{k}, c) = t]\!]$
```

Game G_1 is identical to G_0, except that the CVO is replaced with an alternative implementation. Instead of executing the decryption routine $\hat{m} \leftarrow \text{Dec}(\text{sk}, (c, t))$ to recover the plaintext, the alternative implementation checks previous hash queries in $\mathcal{L}^G$ to recover the plaintext.

Algorithm 5: CVO_1 in G_1

```
Input: $(c \in \mathcal{C}, t \in \mathcal{T})$
1 if $\exists(m, r, k) \in \mathcal{L}^G$ such that $\text{MAC}(k, c) = t$ then
2 |   return 1;
3 end
4 return $\perp$;
```

Consider a single ciphertext query (c, t). If the query is accepted by CVO_1, then this cipehrtext query is honestly generated, so the correctness of PKE and MAC implies that the query will also be accepted by CVO. Similarly, if the query is rejected by CVO, then either the re-encryption does not match, or the MAC does not match; in any case, CVO_1 will also reject the ciphertext query. Therefore, these two implementations disagree if and only if CVO accepts the query but CVO_1 rejects the query.

Since CVO accepts the query, t is a valid tag for c under some MAC key k derived by hashing the decryption $m \leftarrow \text{Dec}(\text{sk}, c)$. Since there is no matching query in $\mathcal{L}^G$, under the random oracle assumption, the MAC key k is indistinguishable from truly random. In other words, for some ciphertext c of the adversary's choosing, t is a forgery under an unknown key. Therefore, for a single ciphertext query, the probability that CVO and CVO_1 disagree is bounded by the advantage of some existential forgery adversary: $P[\text{CVO}(c, t) \neq \text{CVO}_1(c, t)] \leq \epsilon_{\text{MAC}}$.

Game 1 and game 0 behave differently if one or more among all q_V ciphertext validation queries satisfies the condition above, therefore:

$$\epsilon_0 - \epsilon_1 \leq q_V \cdot \epsilon_{\text{MAC}}$$

Game 2 G_2 is identical to G_1, except that the standard PCO is replaced by an alternative implementation PCO_1

Algorithm 6: PCO_1 in G_2

```
Input: $m \in \mathcal{M}, (c \in \mathcal{C}, t \in \mathcal{T})$
1 $(r, k) \leftarrow G(m)$;
2 return $[\![\text{Enc}(\text{pk}, m; r) = c]\!]$ and $[\![\text{MAC}(k, c) = t]\!]$
```

□

4 Experimental results

5 Open questions

Can we get rid of the coin and just let the encryption scheme be probabilistic? Do we need to?