https://doi.org/10.7840/kics.2025.50.2.311

Security Issues of ID-Based on/offline Signcryption Revisited

Jin Wook Byun◆°

ABSTRACT

ID-based offline and online signcryption (OOSE) consists of two phases; offline signcryption (OffSE) and online signcryption (OnSE). Through OffSE phase, senders can pretreat heavy operations, which enables for senders to perform relatively light operations in OnSE phase. In the paper, we raise an access issue between two phases; if a sender A can access OffSE then is it right to permit A to naturally get an access to OnSE? In practice, however, to connect two phases, there must be secure long-term memory spaces in which OffSE secrets are stored. If those memories are corrupted by accident, then anyone can freely proceed with the next OnSE phase. What we claim here is that, for better security, each phase must not affect the security of the other phase under the corruption of each phase. We firstly present these offline and online attacks in OOSE and analyzed them through relevant OOSE results.

Key Words : ID-based Signcryption, on/offline cryptography, memory corruption, on/offline attack, security analysis

Ⅰ. Introduction

An ID-based on/offline signcryption (OOSE) is one of the well-developed cryptography primitives that securely merges three cryptographic features; an ID-based cryptography, a signcryption cryptography, and an on/offline function. First, ID-based cryptography, initially proposed by Shamir[1], aims to use a simple identifier (ID) or an email address to encrypt messages rather than relying on the existing public keys that always need tiresome validation procedure to verify digital certificates. Second, a signcryption (for short, SE) is literally a combination of *signature* and *encrypton* primitives, firstly suggested by Zheng[2], has now become a indispensable primitive that guarantees both *confidentiality* and *authentication*. Lastly, an idea of the offline and online function is to separate one signcryption phase into two phases; offline signcryption and online signcryption. Its purpose is that, through an offline phase, it can save computational costs by pretreating high-cost parts of the whole computations beforehand.

Therefore, an ID-based offline and online signcryption (for short, OOSE) is an enhanced primitive not only solves the issues of managing the certificates, but also efficiently provides confidentiality and authentication properties simultaneously with two separated phases. The study on OOSE had been actively explored for the two decades, but now it seems to have become inactive because many provably secure and efficient schemes have been already presented and their securities are analyzed too enough.

From the viewpoint of OOSE, it is so natural process that original senders who have already produced offline signcryption (for short, OffSE) in an offline phase to subsequently produce online sigcryptions (for short, OnSE) in the online phase. Our motivation, however, starts from breaking this assumption. Instead, we assume that OffSE and OnSE are functionally and physically separated and their working process may not be performed in one specific device at once. That is, after a sender A performs OffSE

※ We thank Young Ah Shin for proofreading of the paper and insightful discussions for future works.

◆° First and Corresponding Author : Pyeongtaek University, Department of Communication and Information; jwbyun@ptu.ac.kr, 정회원

논문번호 : 202407-156-C-RU, Received July 25 2024; Revised August 26, 2024; Accepted October 10, 2024

phase, their outputs c_f are securely managed in the memory space. Then, after a long period of time, the same sender A who brings c_f (*e.g.* through USB) may do OnSE through other remote devices. At this point, a new access issue, "*Is it right for anyone who just brings c_f to freely access* OnSE *phase without being required anything?*", can be raised. To enhance security, each phase must be independent and not affect the security of the other phase under the corruption of each phase.

We observe that, up to now, most schemes do not consider such malicious scenario. In the paper, thus, we firstly raise an access issue that anyone who holds c_f from OffSE can produce a valid online signature without any permission of senders. Furthermore, reversely, anyone who can access final OnSE phases can produce OffSE or OnSE freely. In this paper, we present these offline and online attacks in OOSE and analyzed them through relevant OOSE results. Next, we explain the existing OOSE with security definitions. At last, our observations and its analysis based on new attacks are newly presented.

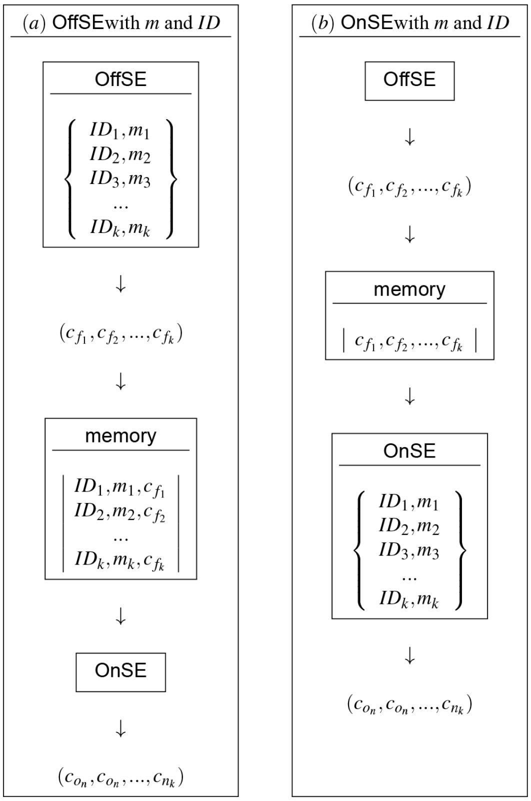
Ⅱ. OOSE Process and Definitions

2.1 On/offline signcryption process

An ID-based OOSE consists of six phases. We suppose that a sender ID_s sends signcryption for m to a receiver ID_r.

- Setup : It generates public parameters such as multiplicative cyclic group, generator, description of hash, and pairing operation e. It also generates a master key *msk* and a public key P_{pub}.
- Extract : On inputs *msk*, *ID*, it generates a secret key d for an identifier *ID*
- Offline Signcrypt (OffSE) : It produces an offline ciphertext c_f on inputs of a sender's secret key d_s and ID_s. It is an intermediate ciphertext that is later delivered to an online signcryption.
- Online Signcrypt (OnSE) : It generates a final signcrypt c_n on inputs of c_f and ID_r.
- DeSigncrypt : On inputs c_n , ID_s, ID_r, it decrypts c_n with the receiver's secret key d_r and outputs message m and signature σ . If c_n is not valid, it outputs a message of invalid
- Verification : It verifies signature σ for m using the sender's ID_s and public values, and then outputs its result (valid or invalid).

Originally, in the field of OOSE, OffSE takes the message m and the receiver's ID_r as inputs and outputs intermediate c_f . Then OnSE takes c_f and produces c_n as a final signcrypt. Recently, however, a new idea, which the message m and *IDr* are later given to OnSE (not to OffSE), has been suggested. This approach enables the sender to more flexibly prepare OffSE operations without the information of receiver's ID_r and message m. That is, senders can prepare as many OffSE outputs as possible in offline phases, which the sender can use for any *ID* and messages. This



♦ ↓ denotes a transmission from a upper part to a lower part over wireless or wire channel. |a| implies that a content is being stored in a storage for long-term period while {a} means that a is used for temporary input for process, not just for storage purpose.

Fig. 1. Comparison of two OOSE paradigms

procedure is definitely more convenient than existing approach that OffSE requires them all in advance.

As compared in Fig 1, the left side (a) shows the previous approach that OffSE takes *ID* and *m* while the right side (b) shows the new approach that OnSE takes *ID* and *m*. One best advantage of approach (*b*), the sender can merge any offline signcrypts c_{fi}, $1 \leq i \leq k$ with any identifier, message *IDi* , *mi* and conveniently make a final signcrypt c_{oi}, $1 \leq i \leq k$.

2.2 OOSE security definition

OOSE guarantees two formal security properties; message confidentiality and unforgeability, as follows. Confidentiality. An ID-based OOSE is secure against chosen ciphertext attack if no PPT adversary $\mathscr{A}$ gain advantage with non-negligible probability through the following experimental game in which a challenger *C* allows an adversary $\mathscr{A}$ to ask queries defined below and measure its advantage on unforgeability and confidentiality.

- By running Setup phase, $\mathscr{C}$ first obtains *msk* and then provides $\mathscr{A}$ with pubic parameters. $\mathscr{C}$ allows for $\mathscr{A}$ to ask the following queries. The queries can be adaptively made depending on the results of previous queries.
  - Extract : For any *ID*, $\mathscr{A}$ can ask Extract query then obtain a secret key *d* for *ID*.
  - Signcrypt : For ID_s, ID_r, and *m*, $\mathscr{A}$ can ask Signcrypt query then obtain signcryption ciphertext c_n.
  - DeSigncrypt : For c_n, ID_s, ID_r, $\mathscr{A}$ can ask DeSigncrypt query then obtain a message *m* and its signature σ, if c_n is a valid signcryption. Otherwise, $\mathscr{A}$ obtains an invalid notification.
- After queries, $\mathscr{A}$ comes up with two target messages m_0 , m_1, two target identities ID_s, ID_r to $\mathscr{C}$.
- Then $\mathscr{C}$ selects a random bit $b \in \{0, 1\}$ and makes a target ciphertext c_b^* for m_b, ID_s, ID_r.
- Finally, $\mathscr{A}$ guesses the bit b' for *b*. If $b' = b$ then we define $\mathscr{A}$ wins the game. $\mathscr{A}$'s advantage to win is defined as

$$Adv(\mathscr{A}) = \left|Pr[b' = b] - \frac{1}{2}\right| \qquad (1)$$

- Query restriction : To avoid the cases that $\mathscr{A}$ win easily, $\mathscr{A}$ is neither able to ask DeSigncrypt query for c_n^*, ID_s, ID_r nor Extract query for ID_r.

Unforgeability. An ID-based OOSE is existentially unforgeable against chosen message attack if no PPT adversary $\mathscr{A}$ gain advantage with non-negligible probability through the following game.

- As in Definition 1, $\mathscr{C}$ obtains *msk* from running Setup phase. And, $\mathscr{C}$ provides $\mathscr{A}$ with pubic parameters.
- $\mathscr{C}$ allows for $\mathscr{A}$ to adaptively ask the queries as defined in Definition 1.
- After queries, $\mathscr{A}$ finally outputs c_n^* for ID_s, ID_r. If DeSigncrypt for c_n^*, ID_s, ID_r is a valid then we define that $\mathscr{A}$ wins the game. $\mathscr{A}$'s advantage is the probability of winning the game.
- Query restriction : To avoid the trivial cases, $\mathscr{A}$ is neither able to ask DeSigncrypt query that produces *m*, σ nor Extract query for ID_s.

Ⅲ. Observations of the Existing Schemes

3.1 Structural difference

Basically, OOSE (either type (a) or (b) in Fig. 1) is different with the original SE in a way that OOSE is designed to separate a signcryption step into two (on/ off) phases, due to this, it certainly requires long-term memory space to connect the two phases. In other words, for the later computation in OnSE, the outputs of the first OffSE must be stored in a secure memory space. Despite their difference, one notable thing is that many OOSE schemes so far follow the security model of the existing ID-based SE schemes. In OOSE, one can easily observe that the number of types of behaviors for $\mathscr{A}$ can be more variant than the previous SE. Referring to OOSE definitions, nonetheless, the experiment permits a Signcrypt query, despite being OOSE with on/offline phases. During two phases, for instance, $\mathscr{A}$ can ask either OffSE or OnSE separately, not just one

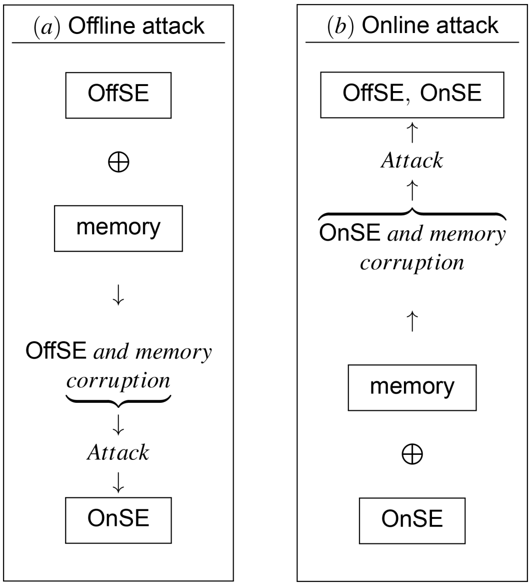


Fig. 2. Online and Offline attacks

Signcrypt query in SE. Also, $\mathscr{A}$ may corrupt memory space to gain much advantage in the game. In Fig 2, two new types of attacks are illustrated with these single phase attacks.

3.2 New attacks

Based on the observations above, our new attacks assume that OffSE and OnSE are physically disparate, thus, there arises an access issue between OffSE and OnSE. That is, if a sender $\mathscr{A}$ can access OffSE then is it right to permit A to get an access to OnSE as a matter of course? In the existing OOSE, it is assumed that A, which has access to OffSE and memory, can naturally do perform OnSE. To be more specific, OffSE requires sender's secret key while OnSE just needs random values (internally generated or obtained from memory) without requiring any secret long-term values. If we suppose an adversary $\mathscr{A}$ who corrupts c_f of OffSE and secret values in the memory space, then, using them, $\mathscr{A}$ can produce a final signcryption c_n of OnSE. As illustrated in Fig 2, we simply define this type of attack as an offline attack (for short, $\Delta_{off\rightarrow on}$ in Table 1). Reversely, $\mathscr{A}$ can use the existing c_n and c_f in the memory spaces to produce a secret d used for OffSE. It makes it possible for $\mathscr{A}$ to forge every next c_f, c_n. We define this type of attack as an online attack (for short, $\Delta_{on\rightarrow off}$ in Table 1).

3.3 Analysis of the schemes and discussions

3.3.1 Analysis

One remarkable result analyzed in Table 1 is that most OOSE schemes do not guarantee the security against the offline attack. The reason is not from any security fault but it is from the existing OOSE definition, such that only their OffSEs requires secrets for an input. For instance, as in the algorithms (Table 1), SHMS scheme[3], the secret d is used for making S in OffSE while OnSE simply perform basic operations for signcryption using c_f values. Thus, any adversary can easily produce any further valid signcryptions without any consent if once c_f is corrupted.

Interestingly, one result NRKAKY[9] has been presented to use a secret d as an input for OnSE, thus, $\mathscr{A}$ without knowing d should not mount the offline attack using c_f. On the other side, however, in the scheme, once the adversary $\mathscr{A}$ (can be an insider or outsider) obtains an arbitrary output of OnSE and its corresponding c_f in the memory, then $\mathscr{A}$ can produce new c_f, c_n values, as follows.

- Let's suppose $\mathscr{A}$ obtains c_f (U, W, y, k), and its corresponding c_n (h, V, C), for a receiver ID_r.
- Then $\mathscr{A}$ simply computes $(V-W)\cdot h^{-1}$ to get d.
- $\mathscr{A}$ with d can do follow every steps from OffSE to OnSE then make new c_f, c_n for a new message m.

The above reverse scenario relies on the fact that two situations (OnSE and memory) are perfectly stolen by $\mathscr{A}$, which has not been considered in the existing schemes, up to date either. As in Table 1, we have analyzed relevant schemes regarding the offline and online attacks. To the best of our knowledge, there exists no scheme considering two attacks, which implies both $\Delta_{on\rightarrow off}$ and $\Delta_{off\rightarrow on}$ are N.

3.3.2 Observations

- ***Most* OOSE *schemes have been designed for* OffSE *to require a secret key.*** One clear observation is that most schemes of OffSEs only tackle the secret key (not with both phases). That is,

Table 1. Analysis of relevant protocols

Scheme	Setup[1]	OffSE[2]	c_f[3]	OnSE[4]	$\Delta_{off\to on}$[5]	$\Delta_{on\to off}$[6]	Type[7]
SHMS[3] (2008)	$msk = s$ $P_{pub} = s\cdot P$ $d = s\cdot H(ID)$	$x,y \leftarrow \mathsf{R}$ $k = H(e(P_{pub},H(ID_r))^x)$ $k \rightsquigarrow k_1,k_2$ $S = d - x\cdot P_{pub}$ $U = (y-k_1)\cdot P$	S,U,k_2,x,y	$k_3 = H(\boxed{k_2})$ $c = \mathsf{E}_{k_3}(m)$ $r = H(c,\boxed{S},\boxed{U})$ $\sigma = r\cdot\boxed{x}+\boxed{y}$	Y	N	SE
SMS[4] (2008)	d,x $Y = x\cdot P$	$m',r' \leftarrow H_Y$ $\sigma = \mathsf{Sign}_d(h\Vert H_Y)$ $X = y\cdot P,\ y\leftarrow R$ $w = e(y\cdot P_{pub},H(ID_r))$ $y' = H(w)$	m',r',y'	$r = x^{-1}(\boxed{m'}-\boxed{m})+\boxed{r'}$ $c = \mathsf{E}_{y'}(\sigma\Vert ID_s\Vert m\Vert r\Vert H_Y)$	N	Y	SE
LZ[5] (2009)	$msk = s$ $d = \frac{1}{H(ID)+s}\cdot P$	$u,x,a,b \leftarrow \mathsf{R}$ $U = u\cdot P$ $R = e(P,P)^x$ $\beta = H(R)$ $T_1 = a^{-1}\cdot x\cdot P$ $T_2 = x\cdot b\cdot P + x\cdot P_{pub}$	u,x,a,b U,R,β,T_1,T_2	$t_1 = \boxed{a}\cdot(H(ID_R)-\boxed{b})$ $t_2 = H(m,\boxed{\text{OSV}},t_1)\cdot\boxed{x}+\boxed{u}$ $c = \boxed{\beta}\oplus m$	Y	Y	E
SVR[6] (2011)	$msk = s$ $P_{pub} = s\cdot P$ $d = \frac{1}{H(ID)+s}\cdot P$	$\delta,b,x,y,z,r \leftarrow \mathsf{R}$ $U_1 = e(P,P)^r$ $U_2 = y\cdot P$ $U_3 = z\cdot P$ $V = (r+h_2)\cdot d$ $h_2 = H(U_1,U_2,U_3,\delta,ID_s)$ $a = H(\delta,V,ID_s)$ $C_1 = a^{-1}\cdot x\cdot P$ $C_2 = x\cdot b\cdot P + (P_{pub})^x$ $k = H(e(P,P)^x)$	k,w,a,b,y,z C_1,C_2,V U_1,U_2,U_3	$C_3 = \boxed{a}\cdot(ID_r-\boxed{b})$ $C_4 = (m\Vert\boxed{\delta})\oplus\boxed{k}$ $v = \boxed{y}\cdot h+\boxed{z}$ $h = (m,\boxed{\text{OSV}},C_3,C_4,ID_s,ID_r)$	Y	N	SE
LKAT[7] (2012)	$msk = s$ $P_{pub} = s\cdot P$ $d = \frac{1}{H(ID)+s}\cdot P$	$x,\alpha,\beta,\gamma \leftarrow \mathsf{R}$ $r = g^x$ $S' = \alpha\cdot d$ $T' = x\cdot(\beta\cdot P + P_{pub})$ $W = x\cdot\gamma\cdot P$	x,α^{-1},β γ^{-1},r,S' T',W	$C = m\oplus\boxed{H(r)}$ $h = H(m,\boxed{\text{OSV}})$ $\theta = (\boxed{x}+h)\cdot\boxed{\alpha^{-1}}$ $\eta = \boxed{\gamma^{-1}}\cdot(H(ID_r)-\boxed{\beta})$	Y	Y	SE
LBZ[8] (2011)	$msk = s$ $d = g^{\frac{1}{H(ID)+s}}$	$u,x,\alpha,\beta,\gamma,\delta \leftarrow \mathsf{R}$ $U = d\cdot g^{-u}$ $R = e(g^{H(ID_s)}+s,g)^x$ $T_0 = (g^{\alpha H(ID_s)+s\cdot H(ID_s)+\gamma+s^2})^x$ $T_1 = g^{x\cdot\beta^{-1}H(ID_s)}$ $T_2 = g^{s\cdot x\cdot\delta^{-1}}$	$u,x,\alpha,\beta,\gamma,\delta$ U,R,T_0,T_1,T_2	$t_1' = \boxed{\beta}\cdot(H(ID_r)-\boxed{\alpha})$ $t_2' = \delta\cdot(H(ID_r)-\boxed{\gamma})$ $t = h_2\cdot\boxed{x}+\boxed{u}$ $c = h_3\oplus m$ $h_2 = H(m,ID_s,t_1',t_2',\boxed{\text{OSV}})$ $h_3 = H(\boxed{R,T_1,T_2,U})$	Y	N	SE
LKAT[7] (2012)	$msk = s$ $P_{pub} = s\cdot P$ $d = \frac{1}{H(ID)+s}\cdot P$	$x,\alpha,\beta,\gamma \leftarrow \mathsf{R}$ $r = g^x$ $S' = \alpha\cdot d$ $T' = x\cdot(\beta\cdot P + P_{pub})$ $W = x\cdot\gamma\cdot P$	x,α^{-1},β γ^{-1},r,S' T',W	$C = m\oplus\boxed{H(r)}$ $h = H(m,\boxed{\text{OSV}})$ $\theta = (\boxed{x}+h)\cdot\boxed{\alpha^{-1}}$ $\eta = \boxed{\gamma^{-1}}\cdot(H(ID_r)-\boxed{\beta})$	Y	Y	SE
NRKAKY[9] (2021)	$msk = s$ $P_{pub} = s\cdot P$ $d = s\cdot H(ID)$	$x \leftarrow \mathsf{R}$ $U = x\cdot P$ $W = x\cdot P_{pub}$ $y = e(W,H(ID_r))$ $k = H(y)$	U,W,k	$h = H(\boxed{U},m)$ $V = h\cdot d+\boxed{W}$ $C = m\oplus\boxed{k}$	N	Y	SE
LHHW[10] (2021)	$msk = s$ $P_{pub} = s\cdot P_2$ $g = e(P_1,P_{pub})$ $d = \frac{s}{H(ID\Vert p)+s}\cdot P_1$	$r,k \leftarrow \mathsf{R}$ $w = g^r$ $S = (r-k)\cdot d$	r,k,w,S	$h = H(M\Vert\boxed{w},p)$ $\tau = (\boxed{r}-h)(\boxed{r}-\boxed{k})^{-1}$	Y	N	S

† A notation $\boxed{\text{a}}$ does mean that a comes from c_f (the output of OffSE). Setup denotes a process of generating a master key (msk) s and a secret key d with public key P_{pub}. We omit explanation on public parameters for groups, a bilinear pairing e, and elliptic curves for simplicity. Regarding hash functions, we apply the same notation h although schemes use different hash functions. OffSE and OnSE imply real processes of the offline and online phases in OOSE. In the process, $x \leftarrow R$ means x is chosen randomly. Also, OSV notation denotes the values are consist of c_f values (for simplicity, its full messages are not described in the table). $\Delta_{off\to on}$ and $\Delta_{off\to on}$ imply online and online attacks explained section 3.2. Y denotes the attack is possible while N is impossible.

at the first OffSE, it produces authentic data c_f with the secret value, in which a digital signature is applied, then at the second OnSE, c_f is simply merged to make signcryptions with the message m and ID, which only require lightweight operations such as multiplication and addition without making any other random values. This unbalance computation process, in a sense, is certainly an efficient design of OOSE. However, we observe that, regarding most OOSE schemes, if once c_f is corrupted, then all security is not guaranteed, as summarized in Table 1.

- ***Although both phases have its own private keys, the scheme can be insecure against on/offline attacks.*** The simple way to handle these attacks is to design OnSE to take a sender's secret as OffSE does. However, unfortunately, it is never a simple work to securely add sender's secrets into each phases.

For instance, let's see the LKAT algorithm[7] in the table. The scheme takes a secret d in OffSE to make S' while OnSE does perform simple operations using c_f values without any secret. According to our claims, due to d in OffSE, no adversary is supposed to perform OffSE at all. However, an adversary is able to obtain α^{-1} from c_f . This implies that the adversary can capture any past OffSE message S' from c_f and easily compute d.

$$S' \cdot \alpha = \alpha \cdot d \cdot \alpha^{-1} = d$$

The same applies to another scheme SMS[4]. The scheme also requires secrets d and x that are used for each step OffSE and OnSE, respectively to guarantee unforgeability. Although both phases take each secret value, its issue comes from y' in c_f , which is used as a symmetric key to make c. Under the corruption of c_f , any adversary can recover m, which breaks confidentiality.

This analysis reminds us of the fact that each phase can be insecure even though they are designed with secret value at each step. If OffSE produced the related secret value from output c_f , then any adversary with c_f may compute the secret value, which breaks unforgeability and confidentiality of signcryptions. Therefore, how to securely make each secret values and c_f must be a careful consideration considered for constructing a secure OOSE.

- ***Only one scheme SMS[4] is designed with two separate secrets, but it already inherits other security vulnerabilities.*** What we claim here is that, when we design OOSE, the scheme can guarantee much security if their two phases (offline and online) were designed with independent secret values to make its output, which definitely does not affect the security of the other phase, even if each phase corruption happens. Our observation is that there has been one such scheme SMS[4], as analyzed in Table 1. Although SMS[4] does not take into accounts of these attacks in their security model, the scheme SMS were wisely designed with separate private keys in their on/offline phases.

However, other security concern in SMS has been found by Selvi et al.[4] They have shown that an adversary can produce a valid signcryption for a message m, a sender ID_A, and a receiver ID_C through using a valid signcryption for m, a sender ID_A, and a receiver ID_B. Please note that this is not the security breach from on/offline attacks discussed here, but under their security model. To the best of our knowledge, there exists no scheme that is secure against on/offline attacks in recent provably secure schemes.

3.3.3 Other cryptographic primitives

It is worthwhile to see the case of on/offline encryption (for instance, LZ[5] in Table 1) where two phases are not interested in taking their secret inputs since it is a public encryption that must be performed only with public values. Hence, on/offline public encryption does not relate to our security issues.

Another case of on/offline signature (LHHW[10] in Table 1), however, does have concerns with our security issues, since the scheme is a digital signature, and a secret d is used to make S in OffSE. In other words, in OnSE phase, only c_f values are required without any secret value. When we say on/offline phases are secure, normally each phase corruption should not affect other phase's security (forgery). As described in Table 1, when an adversary corrupts c_f

from memory, anyone can perform OnSE, which denotes that an adversary can produce any valid signature. Our result shows that $\Delta_{off \leftarrow on}$ is possible in LHHW while its reverse $\Delta_{off \leftarrow on}$ is impossible due to the secret value *d*.

3.4 Discussion on Countermeasures

Our new assumption here is that an adversary $\mathscr{A}$ is allowed to obtain c_f from memory corruptions. This assumption more empowers the behaviors of the adversary, making the security model much stronger at the same time. For example, queries for corruptions of memory to get c_f must be required in the model in addition to existing Extract query to obtain the secret key.

One generic solution is to apply other existing secure cryptographic primitives (a secure digital signature (Sign), a public key encryption (PE), a symmetric key encryption (SE)) into OOSE. Let's suppose a sender ID_A and a receiver ID_B .

- OffSE : A secure PE with ID_A 's public key can be used for encryption of K where K is a symmetric key for SE. Since the PE with ID_A 's public key can be decrypted with ID_A 's private key, this is exactly a self encryption that only ID_A can decrypt at OnSE. We define the ciphertext of PE as *a*, then, using Sign, we make a signature δ for *a*. δ and *a* are stored to memory for later use.
- OnSE : The sender ID_A verifies δ, if it is valid then decrypts *a* by its own private key and recover K. The sender encrypts a message *m* with K and makes a ciphertext *b*. Lastly, *b* is also signed using Sign then delivered to ID_B with *b*.

This generic approach is similar with PGP protocol for email security, but different in the sense of self encryption from OffSE to OnSE. That is, for message confidentiality, our OffSE phase allows a sender to encrypt symmetric key K with own public key, which the sender later decrypts them in OnSE. Due to the double usage of signatures, it shows low efficiency compared with the existing protocols. An on/offline secure design of OOSE for gaining efficiency remains future work, including the method to design a secure c_f in a security model such that it should not affect confidentiality and unforgeability, even though c_f is corrupted.

Ⅳ. Concluding Remarks

Basically our observations are based on the fact that most OnSE in OOSE do not need the sender's secret as an input while OffSE in OOSE needs the sender's secret. As a result, the risky points arise when OffSE phase is completed, because its output secrets c_f are managed in a long-term memory (not RAM, but flash memory or ROM) for later computations. What we are concerned about is that these long-term memory may become vulnerable by memory leaks or corruptions in any IoT situations. Therefore, we first present the novel offline/online attacks from those vulnerability and analyzed them in Table 1.

References

[1] A. Sharmir, “Identiy-based cryptography and signature schemes,” in *Proc. CRYPTO 84, LNCS*, vol. 196, pp. 47-53, 1984. (https://doi.org/10.1007/3-540-39568-7_5)

[2] Y. Zheng, “Digital signcryption or how to achieve cost(signature encryption) ≫ cost (signature) + cost(encryption),” in *Proc. CRYPTO, LNCS*, pp. 165-179, 1997. (https://doi.org/10.1007/BFb0052234)

[3] D. Sun, X. Huang, Y. Mu, and W. Susilo, “Identity-based on-line/off-line signcryption,” in *Proc. Int. Conf. Netw. and Parallel Comput.,* pp. 34-41, Shanghai, China, 2008. (https://eprint.iacr.org/2010/376)

[4] D. Sun, Y. Mu, and W. Susilo, “A generic construction of identity-based online/ offline signcryption,” in *Proc. IEEE Int. Symp. Parallel and Distrib. Process. with Appl.,* pp. 707-712, Sydney, NSW, Australia, 2008. (https://doi.ieeecomputersociety.org/10.1109/ISPA.2008.16)

[5] J. Z. Joseph and K Liu, “An efficient identity-based online/offline encryption scheme,” in *Proc. ACNS09, LNCS*, vol. 5536,

pp. 156-167, 2009. (https://doi.org/10.1007/978-3-642-01957-9_10)

[6] S. Sharmila Deva Selvi, S. S. Vivek, and C. P. Rangan, "Identity based online/offline encryption and signcryption schemes revisited," in *Proc. InfoSecHiComNet, LNCS*, vol. 7011, pp. 111-127, 2011. (https://doi.org/10.1007/978-3-642-24586-2_11)

[7] F. Li, M. K. Khan, and T. Takagi, "Identity-based online/ offline signcryption for low power devices," *J. Netw. and Computer Appl.*, vol. 35, pp. 340-347, 2012. (https://doi.org/10.1016/j.jnca.2011.08.001)

[8] J. K. Liu, J. Baek, and J. Zhou, "Online/ offline identity-based signcryption revisited," in *Proc. ISC11, LNCS*, vol. 6584, 2011. (https://doi.org/10.1007/978-3-642-21518-6_3)

[9] V. S. Naresh, S. Reddi, S. Kumari, V. V. L. D. Allavarpu, S. Kumar, and M.-H. Yang, "Practical identity based online/off-line signcryption scheme for secure communication in internet of things," *IEEE Access*, vol. 9, pp. 21267-21278, 2021. (https://10.1109/ACCESS.2021.3055148)

[10] J. Lai X. Huang, D. He, and W. Wu, "Provably secure online/offline identity-based signature scheme based on SM9," *The Computer J.*, vol. 65, pp. 1692-1701, 2011. (https://doi.org/10.1093/comjnl/bxab009)

Jin Wook Byun

Feb. 2001 : B.Eng. degree, Department of Computer Science, Korea University, Sejong, Rep. of Korea.

Feb. 2003 : M.Eng. degree, Graduate School of Information and Security, Korea University, Seoul, Rep. of Korea.

Aug. 2006 : Ph.D. degree, Graduate School of Information and Security, Korea University, Seoul, Rep. of Korea.

Mar. 2008~Current : A full-time professor, Dep. of Information and communication, Pyeongtaek University, Pyeongtaek, Rep. of Korea

<Research Interests> security protocol, cryptography, digital signature, keyword search on encrypted database, PUF-based security protocol

[ORCID:0000-0002-5450-3207]