Universally Composable Secure Group Communication

Youliang Tian1,2 and Changgen Peng1,2

1 College of Science, Guizhou University, Guiyang 550025, China
2 Institute of Cryptography and Data Security, Guizhou University, Guiyang 550025, China
(Email: tianyouliang@iie.ac.cn)

Abstract. This paper analyzes group communication within the universally composable framework. We first propose the group communication model, identity-based signcryption model and group key distribution model in the UC framework by designing the ideal functionality Fsagcom, Fidsc and Fgkd, respectively. Then, we construct a UC secure identity-based signcryption protocol πidsc. Moreover, we shows that the identity-based signcryption πidsc securely realizes the ideal functionality Fidsc if and only if the corresponding protocol ID-SC is secure. Finally, based on the identity-based protocol, we propose a group communication scheme πsagcom, which can securely realizes the ideal functionality Fsagcom in the (Fidsc,Fgkd)-hybrid model.

Keywords: Universally composable secure, Secure group communication, Identity-based signcryption, Many-to-many group communication

1 Introduction

As is known to all, multicasting is increasingly used as an efficient communication mechanism for delivering data to multiple receivers in one-to-many or many-to-many group-oriented applications in the Internet or public broadcasting. However, the lack of security in the multicast communication model obstructs the effective and large scale deployment of multi-party applications. Much research has been done into being traditionally defined as stand-alone scheme problems without giving much attention to more complex execution environments to to multicast group communications. Therefore, it is extremely interesting to study multicast group communication scheme within the universally composable framework.

Related work Fiat and Naor [2] introduced broadcast encryption, which provided a method of securely broadcasting key information such that only a privileged set of users can decrypt the information while a coalition of up to k other users cannot know anything. Wang and Wu [4] proposed an authenticated identity-based multicast scheme from bilinear pairing, where security is also provided. However, Lin et al. [5] found that [4] is not secure against the insider forgery attack. Mu et al. [6] proposed another identity-based authenticated broadcast encryp-

tion scheme, which allows each sender to dynamically broadcast messages to its group members using a polynomial function constructed with secret keys of the members. Hur et al. [7] proposed an authenticated group communication scheme which is secure against an adaptive chosen ciphertext attack using identity-based signcryption. And solved the sender authentication problem by using an identity-based signcryption framework.

However, the above mentioned schemes do not study their security in the universally composable framework. Thus these schemes could not satisfy universally composable secure.

The UC framework The universally composable (UC) framework [1] for analyzing security of cryptographic protocols provides very strong security guarantees. In particular, a protocol proven secure in this framework is guaranteed to maintain its security even when it is run concurrently with other protocols, or when it is used as a component of a large protocol. Ideal functionality is an extremely important security concept in the UC framework; it serves as an uncorruptable trusted party and can realize the specific task of carrying out the protocol.

At present, most basic ideal functionalities have already been defined, such as the message authentication functionality Fauth, the key-exchange functionality Fke, the public-key
In this paper, we investigate secure authenticated group communication based on identity-based signcryption. Our contributions are shown below. Our contributions are shown below.

1. We propose a universally composable group communication model including the ideal functionalities identity-based signcryption $F_{IDSC}$, group key distribution $F_{GKD}$ and secure authenticated group communication $F_{SAGCOM}$.

2. According to $F_{IDSC}$, we design a protocol $\pi_{IDSC}$ to realize the identity-based signcryption functionality in the hybrid model. Meanwhile, we show that $\pi_{IDSC}$ securely realizes $F_{IDSC}$ if and only if IDSC is secure with respect to both IND-IDSC-CCA2 and EXT-IDSC-CMA.

3. Based on $F_{IDSC}$ and $F_{GKD}$, we construct a scheme $\pi_{SAGCOM}$ to realize the secure authenticated group communication functionality $F_{SAGCOM}$ in the $(F_{IDSC}, F_{GKD})$-hybrid model.

The rest of this paper is organized as follows: Section 2 gives a brief introduction to the UC framework and identity-based signcryption. In section 3, we propose the ideal functionalities of identity-based signcryption, group key distribution and secure authenticated group communication. An identity-based signcryption protocol is proposed in section 4, while in section 5, we present the secure authenticated group scheme. Finally, we conclude our results in section 6.

2 Preliminaries

In this section, we briefly review the framework of UC [1] and identity-based signcryption scheme [3].

2.1 The UC framework

Firstly, the process of executing a protocol in the presence of a real-world adversary is formalized in the UC framework. Secondly, an ideal process for carrying out the task at hand is formalized. In the ideal process, the parties do not communicate with each other. Instead they have access to an ideal functionality, which is essentially an incorruptible “trusted party” that is programmed to capture the desired functionality of the task at hand.

**Definition 1.** (UC emulation) We say that a protocol $\pi$ UC-realizes an ideal functionality $F$ if for any real-world adversary $A$, there exists an ideal adversary $S$ such that for any environment $Z$, the probability that $Z$ is able to distinguish between an interaction with $A$ and real parties running protocol $\pi$ and an interaction with $S$ and dummy parties accessing $F$ in the ideal process is at most a negligible probability, i.e. $\Pr[REAL_{\pi,A,Z} \approx IDEAL_{F,S,Z}] \approx 0$.

**Theorem 1.** (Composition Theorem) Let $\rho$ be a protocol that securely realizes the ideal functionality $F$, and let $\pi$ be a protocol in the $F$-hybrid model. We say that $\pi^{\rho/F}$ with the ideal functionality $F$ which is replaced by $\rho$, UC-realizes $\pi$. In particular, if $\pi$ securely realizes the ideal functionality $\xi$ in the $F$-hybrid model, then $\pi^{\rho/F}$ securely realizes $\xi$ from scratch.

According to the Composition Theorem, a large protocol can be constructed by using some sub-protocols which are proven secure in the UC framework. This is very important since a complex but secure system can usually be divided into a number of sub-systems, each one performing a specific task securely.
2.2 Identity-based signcryption

Our definition of an identity-based signcryption scheme is identical to the one given in [3].

IDSC scheme An identity-based signcryption scheme consists of Setup, Extract, Signcrypt, and Unsigncrypt algorithms. The functions of these algorithms are described below.

1. Setup The setup algorithm produces global public parameters, the master secret key s, and the master public key on the input of security parameter 1^k.
2. Extract The Extract algorithm outputs a secret key of a user on the input of the master secret key and the public identity of the user.
3. Signcrypt The Signcrypt algorithm produces a ciphertext σ, which is the signcryption of m with signcrypter’s secret key and the public identity of the receiver.
4. Unsigncrypt The receiver uses the Unsigncrypt algorithm to unsigncrypt the received ciphertext σ with its own secret key and the identity of the signcrypter recover the corresponding plaintext m, and the receiver can verifies the signcryption by checking whether σ is the signcryption of m on using the Unsigncrypt algorithm. If it is, it outputs 1; and it outputs ⊥ otherwise.

Security requirement In the identity-based signcryption scheme, confidentiality and non-repudiation are always required. We need two experiments described as followings.

\[ \text{Exp}^{\text{IND-CCA2}}_{\text{IDSC,A}}(1^k) \]
\[ (s, P_{\text{pub}}) \leftarrow \text{Setup}(1^k) \]
\[ (sk_s, ID_s) \leftarrow \text{Extract}(s, P_{\text{pub}}, ID_s) \]
\[ (sk_r, ID_r) \leftarrow \text{Extract}(s, P_{\text{pub}}, ID_r) \]
\[ (m_0, m_1, \text{state}) \leftarrow A_1^{O_e, O_s, O_u}(ID_s, ID_r) \]
\[ b \leftarrow \{0, 1\} \]
\[ \sigma \leftarrow \text{Signcrypt}(ID_s, ID_r, P_{\text{pub}}, m_b) \]
\[ b' \leftarrow A_2^{O_e, O_s, O_u}(ID_s, ID_r, m_0, m_1, \sigma, \text{state}) \]

If \( b' \neq b \) then return 1, otherwise return 0.

\[ \text{Exp}^{\text{EXT-CMA}}_{\text{IDSC,A}}(1^k) \]
\[ (s, P_{\text{pub}}) \leftarrow \text{Setup}(1^k) \]
\[ (sk_s, ID_s) \leftarrow \text{Extract}(s, P_{\text{pub}}, ID_s) \]
\[ (sk_r, ID_r) \leftarrow \text{Extract}(s, P_{\text{pub}}, ID_r) \]
\[ \sigma \leftarrow A_0^{O_e, O_s, O_u}(ID_s, ID_r) \]

If \( \text{Unsigncrypt}(ID_s, ID_r, \sigma) \neq \bot \) then return 1, otherwise return 0.

Here \( O_e, O_s, O_u \) represent Extract oracle, Signcrypt oracle and Unsigncrypt oracle, respectively. The first experiment concerns privacy of messages, and adapts the notion IND-IDSC-CCA2 from public key encryption. \( \mathcal{A} \) is said to win if the experiment returns 1 with non-negligible advantage. The second experiment concerns unforgeability of messages, and adapts the notion EXT-IDSC-CMA from digital signatures. \( \mathcal{A} \) is said to win if the experiment returns 1 with non-negligible advantage.

Next we define the advantage of \( \mathcal{A} \) in breaking IDSC with respect to IND-IDSC-CCA2 as

\[ \text{Adv}^{\text{IND-CCA2}}_{\text{IDSC,A}}(1^k) = |2Pr[\text{Exp}^{\text{IND-CCA2}}_{\text{IDSC,A}}(1^k) = 1] - 1| \]

The scheme IDSC is said to be secure with respect to IND-IDSC-CCA2 if the advantage \( \text{Adv}^{\text{IND-CCA2}}_{\text{IDSC,A}}(1^k) \) is negligible in \( \epsilon \), whenever \( \mathcal{A} \)'s runtime and number of oracle queries are polynomially bounded in \( \epsilon \).

We define the success rate of \( \mathcal{A} \) in breaking IDSC with respect to EXT-IDSC-CMA as

\[ \text{Succ}^{\text{EXT-CMA}}_{\text{IDSC,A}}(1^k) = Pr[\text{Exp}^{\text{EXT-CMA}}_{\text{IDSC,A}}(1^k) = 1] \]

The scheme IDSC is said to be secure with respect to EXT-IDSC-CMA if the success rate \( \text{Succ}^{\text{EXT-CMA}}_{\text{IDSC,A}}(1^k) \) is negligible in \( \epsilon \), whenever \( \mathcal{A} \)'s runtime and number of oracle queries are polynomially bounded in \( \epsilon \).

3 Ideal functionalities

In this section, we mainly consider the ideal procedures for a secure group communication and an identity-based signcryption scheme. Then their functionalities are presented based secure requirements of these protocols.
3.1 Functionality, $\mathcal{F}_{\text{SAGCOM}}$

A secure authenticated group communication means that a party $R$ or a group $G$ will receive a message $m$ from some parties $S$ only if $S$ has sent the message $m$ to $R$ or $G$, and in addition adversary and non-group members have no access to the contents of the transmitted message. Obviously, the main security properties of a secure group communication are secrecy and authenticity. Specifically, the security requirements are considered to be of message confidentiality, ciphertext authentication, and signature non-repudiation.

Confidentiality  Message confidentiality prevents outsiders or non-group members from decrypting the group message. It allows the communicating parties to preserve the secrecy of their communications.

Authentication  Ciphertext authentication allows only the intended legitimate recipient to be convinced that the message was encrypted by the same person who signed it. That is, an outside adversary cannot re-encrypt a signed message of the sender throughout the transmission. This implies ciphertext integrity.

Non-repudiation  Signature non-repudiation prevents the sender of a signed message from disavowing its signature. It can be also verified by only the intended recipient of the signature.

In the present formalization, protocols that assume ideally secure group communication can be cast as hybrid protocol with ideal access to an “authenticated message transmission functionality” and “secure message functionality”. This functionality, denoted $\mathcal{F}_{\text{SAGCOM}}$, is presented in Figure 1.

3.2 Functionality, $\mathcal{F}_{\text{IDSC}}$

Here, we define an ideal functionality of identity-based signcryption, based on the security requirements given in section 2.2. The identity-based signcryption functionality, denoted $\mathcal{F}_{\text{IDSC}}$, is defined in Figure 2.

3.3 Functionality, $\mathcal{F}_{\text{GKD}}$

In this section, we formulate an ideal functionality, $\mathcal{F}_{\text{GKD}}$ (shown in Figure 3), to provide a trusted “the group key distribution service”.

4 Securely realizing $\mathcal{F}_{\text{IDSC}}$

Here we present a universally composable identity-based signcryption scheme $\pi_{\text{IDSC}}$ and its proof of the UC-security in the hybrid model.

4.1 Protocol, $\pi_{\text{IDSC}}$

The protocol $\pi_{\text{IDSC}}$ given in Figure 4 is constructed in a natural way from the identity-based signcryption scheme IDSC.

4.2 Security proof of $\pi_{\text{IDSC}}$

Theorem 2. Let IDSC be an identity-based signcryption scheme, $\pi_{\text{IDSC}}$ securely realizes $\mathcal{F}_{\text{IDSC}}$ if and only if IDSC is secure with respect to both IND-IDSC-CCA2 and EXT-IDSC-CMA.

Proof. $\Rightarrow$ (Reduction to absurdity) Let $\pi_{\text{IDSC}}$ securely realizes $\mathcal{F}_{\text{IDSC}}$, but IDSC is not secure with respect to both IND-IDSC-CCA2 and EXT-IDSC-CMA. Since IDSC is not both IND-IDSC-CCA2 and EXT-IDSC-CMA, then we construct an environment $Z$ and a real-world adversary $A$ such that for any ideal adversary $S$, the environment $Z$ can distinguish between an interaction with $A$ and real parties running protocol $\pi_{\text{IDSC}}$ and an interaction with $S$ and dummy parties accessing $\mathcal{F}_{\text{IDSC}}$ in the ideal process.

1. Suppose IDSC is not IND-IDSC-CCA2, i.e. there exists an adversary $B$ who has non-negligible advantage $\text{adv}^{\text{IND-CCA2}}_{\text{IDSC},A}(1^k)$ in breaking IDSC. The system description follows: In the beginning of the experiment $\text{Exp}^{\text{IND-CCA2}}_{\text{IDSC},A}(1^k)$, the adversary $B$ is given two public identities $ID_s$ and $ID_r$ belonging to the target sender and the target receiver, respectively. $B$ is composed of a find-stage algorithm $B_1$ and a guess-stage algorithm $B_2$. $B_1$ finds two messages $m_0$ and $m_1$ of the same length, while $B_2$ is given a challenge ciphertext $\sigma$ and guesses the bit $b \in \{0, 1\}$ correctly with probability $1/2 + \epsilon$. Environment $Z$ invokes an instance of $\mathcal{F}_{\text{IDSC}}$, and proceeds as follows, in a network of a trusted KGC and two uncorrupted parties $S, R$.

(1) Initially, $Z$ activates $KGC$ with input $(\text{Setup}, \text{sid})$ for $\text{sid} = (\text{KGC, sid})$, obtains system public key $PK_s$ and hands $PK_s$ to $B$.

(2) Next $Z$ activates the parties $S$ and $R$ with input $(\text{Extract}, \text{sid})$ for $\text{sid} = ([S, R], \text{sid})$,}
Functionality $\mathcal{F}_{SAGCOM}$

$\mathcal{F}_{SAGCOM}$ proceeds as follows, when parameterized by leakage function $l : \{0,1\}^* \rightarrow \{0,1\}^*$; and $S$ and $R$ are the sets of some parties.

Upon receiving an input $(\text{Send}, \text{sid}, T, m)$ from some parties $S_i$, where $S_i \in B, B \subseteq S$ and $T \subseteq R$, if $\text{sid} = (B, T, \text{sid}')$ for some sets of senders $S$, then:

- Send $(\text{Send}, \text{sid}, T, l(m))$ to the adversary.
- Generate a private delayed output $(\text{Send}, \text{sid}, m)$ to $R_i$, for all $R_i \in T$.
- Record $B$ sends $m$ to $T$ and halts.
- Else ignore the input.

Upon receiving an input $(\text{Receive}, \text{sid}, B, m)$ from parties $R_i$, if $\text{sid} = (T, B, \text{sid}')$ for some sets of receivers $R$, then:

- Send $(\text{Receive}, \text{sid}, B, l(m))$ to the adversary.
- Generate a public output $(\text{Received}, \text{sid}, \text{ok})$ to $S_i$, for all $S_i \in B$.
- Record $T$ receives $m$ from $B$ and halts.
- Else ignore the input.

Upon receiving $(\text{Corrupt}, \text{sid}, P)$ from the adversary, where $P \in B \cup T$. Then:

- Disclose $m$ to the adversary and record $P$ is corrupted.
- If the adversary provides a value $m'$, and $P \in S$, and no output has been yet written to the receiver, then output $(\text{Send}, \text{sid}, m')$ to the receiver, record it, and halt.

Fig. 1. The secure authenticated group communication functionality, $\mathcal{F}_{SAGCOM}$

obtains their identity $ID_S$ and $ID_r$, and
hands $ID_s$ and $ID_t$ to $B$.

3. When $B$ submits a message $m'$ and two distinct identities $S'$ and $R'$ (one
is a sender identity and the other is a
receiver identity), $Z$ activates $S'$ with
$(\text{Signcrypt}, \text{sid}, ID_{r'}, ID_{r'}, PK_{s'}, m')$, 
obeys a ciphertext $\sigma'$ and hand $\sigma'$ to $B$.

4. When $B$ submits a ciphertext $\sigma'$ and an
identity $R'$ of the receiver, $Z$ activates $R'$ with
$(\text{Unsigncrypt}, \text{sid}, ID_{r'}, ID_{r'}, PK_{s'}, \sigma')$, 
obeys a plaintext $m'$ and hand $m'$ to $B$.

5. When $B$ submits an identity $ID_t$, $Z$ activates $KGC$ with $(\text{Extract}, \text{sid})$, obtains the
 corresponding private key $sk_t$ and hands $sk_t$ to $B$.

6. When $B$ generates the two test plaintexts
$(m_0, m_1)$, $Z$ chooses $b \overset{R}{\leftarrow} \{0,1\}$, activates $S$ with
$(\text{Signcrypt}, \text{sid}, ID_{r'}, PK_{s'}, m_b)$, 
obeys a ciphertext $\sigma^*$, and hangs $\sigma^*$ to $B$ as the
text ciphertext.

7. When $B$ returns a guess $b' \in \{0,1\}$, $Z$
outputs $b \oplus b'$ and halts.

Analyzing $Z$, notice that if interacts with the
adversary $A$ and parties running $\pi_{IDSC}$, then
the instance of $B$ within $Z$ sees in fact a $C$-
CA2 interaction with protocol IDSC. Thus,
in this case $b' = b$ with probability at least
$1/2 + \epsilon$. In the contrast, when $Z$ interacts with
the ideal protocol for $\mathcal{F}_{IDSC}$ and any adver-
sary, the view of the instance of $B$ within $Z$
is statistically independent of $b$, thus in this
case $b' = b$ with probability exactly one half.

2. Suppose IDSC is not $\text{EXT-IDSC-CMA}$, i.e. the success rate $\text{Succ}_{\text{IDSC,A}}^{\text{EXT-CMA}}$ of the experiment $\text{Exp}_{\text{IDSC,A}}^{\text{EXT-CMA}}(1^k)$ is non-
negligible. We construct a simulated instance of $B$, where $B$ is an adversary of forgeable
signcryption with non-negligible probability.

(1) Initially, $Z$ activates $B$. 
(2) Next, $Z$ activates $KGC$ with input
$(\text{Setup}, \text{sid})$ for $\text{sid} = (KGC, \text{sid}')$, obtains
system public key $PK_s$ and hands $PK_s$ to $B$.

(3) When $B$ asks an identity $ID_t$, $Z$ activates
$KGC$ with $(\text{Extract}, \text{sid})$, obtains the corre-
sponding private key $sk_t$ and hands $sk_t$ to $B$.

(4) When $B$ submits a message $m'$ and
two distinct identities $S'$ and $R'$ (one
is a sender identity and the other is a
receiver identity), $Z$ activates $S'$ with
### Functionality $\mathcal{F}_{\text{IDSC}}$

$\mathcal{F}_{\text{IDSC}}$ proceeds as follows, with parties $P_1, \ldots, P_n$, $KGC$ and an ideal adversary $S$.

Upon receiving a message from a corrupted party, $\mathcal{F}_{\text{IDSC}}$ forwards the message to $S$, and when $S$ replies to this message, $\mathcal{F}_{\text{IDSC}}$ forwards the reply to the corrupted party.

#### IDSC.Setup

Upon receiving the message $(\text{IDSC.Setup}, sid)$ from a KGC:

- Send $(\text{IDSC.Setup}, sid, KGC)$ to $S$.
- Upon receiving $(\text{IDSC.Setup}, sid, PK_s)$ from $S$, output $(\text{IDSC.Setup}, sid, PK_s)$ to the KGC.
- Record $(KGC, PK_s)$ and label it ‘fresh’.

#### IDSC.Extract

Upon receiving the first message $(\text{IDSC.Extract}, sid, PK'_s)$ from some party $P_i$, send $(\text{IDSC.Extract}, sid, ID_i, PK'_s)$ to $S$, where $ID_i$ is the identity of $P_i$.

Upon receiving $(\text{IDSC.Received}, sid)$, send $(\text{IDSC.Extract}, sid, ID_i, PK'_s)$ to $P_i$ and KGC.

#### IDSC.Signcrypt

Upon receiving $(\text{IDSC.Signcrypt}, sid, ID_r, PK_s, m)$ from $P_i$, do:

- If $ID_r = ID_j$ for some $j$, $PK_s = PK_l$ for some $l$ and $KGC$ are uncorrupted, then send $(\text{IDSC.Signcrypt}, sid, ID_r, PK_s)$ to $S$.
- Otherwise send $(\text{IDSC.Signcrypt}, sid, ID_r, PK_s, m)$ to $S$.

Upon receiving $(\text{IDSC.Signcrypt.Ciphertext}, sid, ID_r, PK_s, \sigma)$ from $S$, do:

- If there is no recorded entry $(ID_r, ID_s, PK_s, m', \sigma)$ for any $m'$, output $(\text{IDSC.Signcrypt.Ciphertext}, sid, ID_r, PK_s, m, \sigma)$ to $P_i$.
- If $ID_r = ID_j$ for some $j$, $PK_s = PK_l$ for some $l$ and $KGC$ are uncorrupted, then record the entry $(ID_r, ID_s, PK_s, m', \sigma)$.

#### IDSC.Unsigncrypt

Upon receiving $(\text{IDSC.Unsigncrypt}, sid, ID_s, PK_s, \sigma)$ from $P_j$, do:

- Send $(\text{IDSC.Unsigncrypt}, sid, ID_s, ID_j, PK_s, \sigma)$ to $S$.
- Upon receiving $(\text{IDSC.Unsigncrypt.Plaintext}, sid, ID_s, ID_j, PK_s, m', \bot, \sigma)$ from $S$, continue.
- If an entry $(ID_s, ID_j, PK_s, m, \sigma)$ is record, then output $(\text{IDSC.Unsigncrypt.Plaintext}, sid, ID_s, ID_j, PK_s, m, \sigma)$ to $P_j$.
- Otherwise, if $ID_s = ID_i$ for some $i$ and $P_i$ and $PK_s = PK_l$ for some $l$ and $KGC$ are uncorrupted, then output $(\text{IDSC.Unsigncrypt.Plaintext}, sid, ID_s, ID_j, PK_s, \bot, \sigma)$ to $P_j$.
- Otherwise, output $(\text{IDSC.Unsigncrypt.Plaintext}, sid, ID_s, ID_j, PK_s, m'/\bot, \sigma)$ to $P_j$.

---

**Fig. 2.** The identity-based signcryption functionality, $\mathcal{F}_{\text{IDSC}}$
Functionality $F_{GKD}$

$F_{GKD}$ proceeds as follows, with parties $P = \{P_1, \cdots, P_n\}$, KGC and an adversary $S$.

Upon receiving an input $(GKD.Distribute, sid, s)$ from KGC, where $T \in 2^P$ is an access structure and $s$ is the group key, do:

- If there exists $sid = (KGC, T, sid')$, then:
  1. send $(GKD.Distribute, sid, |s|)$ to $S$.
  2. generate a private delayed output $(GKD.Distributed, sid, s_i)$ to $P_i$, for all $P_i \in P$.
  3. record $(sid, s)$ and halt.
- Otherwise halt.
- Once $(sid, s)$ is recorded, ignore any subsequent distribute inputs.

Upon receiving an input $(GKD.Recover, sid)$ from $P_i$, do:

- Add $P_i$ to a set $T$ (initially $T := \emptyset$).
- If there is a set $T \in T$ and there is a recorded distributed group key $s$, then:
  1. send $(GKD.Recover, sid, |s|)$ to $S$.
  2. generate a private delayed output $(GKD.Recovered, sid, s)$ to the parties in $T$.
  3. record $(sid, s)$ and halt.
- Else halt.

Upon receiving a message $(Corrupt, sid, P_i)$ from the adversary $S$, do:

- Add $P_i$ to a set $C$ of the corrupted (initially $C := \emptyset$).
- If the adversary $S$ provides an invalid $s'$ and $(GKD.Distributed, sid, s_i)$ was not yet written on the tape of any uncorrupt party in $T$, then change the recorded value $(sid, s')$.
- If the adversary $S$ provides an invalid $s'$ and $(GKD.Recovered, sid, s)$ was not yet written on the tape of any uncorrupt party in $T$, then change the recorded value $(sid, s')$. Else hands $(GKD.Recovered, sid, s)$ to the adversary $S$.
- Else halt.

Fig. 3. The group key distribution functionality, $F_{GKD}$
Protocol $\pi_{IDSC}$

$\pi_{IDSC}$ proceeds as follows, with parties $P_1, \ldots, P_n$, KGC and an adversary $A$.

**IDSC.Setup**

Upon the first input $(IDSC.Setup, sid)$, KGC runs the algorithms Setup. The KGC obtains a master key pair $(s, P_{pub})$, and outputs $(IDSC.Setup, sid, (s, P_{pub}))$.

**IDSC.Extract**

Upon the first input $(IDSC.Extract, sid, ID)$, $P_i$ runs the algorithms Extract. $P_i$ obtains his private key $sk_i$, and outputs $(IDSC.Extract, sid, (sk_i, ID))$.

**IDSC.Signcrypt**

Upon input $(IDSC.Signcrypt, sid, P_{pub}, ID_r, m)$, $P_i$ obtains $\sigma = Signcrypt(sk_i, P_{pub}, ID_r, m)$ and outputs $(IDSC.Signcrypt, Ciphertext, sid, ID_r, m, \sigma)$.

**IDSC.Unsigncrypt**

Upon input $(IDSC.Unsigncrypt, sid, P_{pub}, ID_s, P_K, \sigma)$, $P_i$ obtains $m/\perp = Unsigncrypt(ID_s, P_{pub}, sk_i, \sigma)$ and outputs $(IDSC.Unsigncrypt, Plaintext, sid, m/\perp, \sigma)$.

Fig. 4. The identity-based signcryption protocol, $\pi_{IDSC}$

(5) If $m*$ is not ‘fresh’, then $Z$ outputs 0. Otherwise outputs whatever $R$'s outputs.

Obviously, when $Z$ interacts with the adversary $A$ and parties running $\pi_{IDSC}$, $Z$ outputs 1 with non-negligible probability since $B$ can forge a valid signcryption. However, when $Z$ interacts with the adversary $S$ and parties running $F_{IDSC}$, $Z$ can not outputs 1.

The above analysis shows that $\pi_{IDSC}$ securely realizes $F_{IDSC} \Rightarrow$ IDSC is secure with respect to both IND-IDSC-CCA2 and EXT-IDSC-CMA.

$\Leftarrow$ It remains to show that if IDSC is secure with respect to both IND-IDSC-CCA2 and EXT-IDSC-CMA, then $\pi_{IDSC}$ securely realizes $F_{IDSC}$. We show that $\pi_{IDSC}$ securely realizes $F_{IDSC}$. Using the equivalent notion of security with respect to the dummy adversary, we construct an ideal-process adversary $S$ such that no environment $Z$ can tell with non-negligible probability whether it interacts with $F_{IDSC}$ and $S$ or with parties running $\pi_{IDSC}$ and the dummy adversary $D$.

Assume for contradiction there is an environment $Z$ that distinguishes between the real and ideal interactions. We use $Z$ construct an adversary $B$ that break the IND-IDSC-CCA2 and EXT-IDSC-CMA security of the identity-based signcryption scheme IDSC.

Adversary $B$ proceeds as follows, given a extract algorithm $Extract$, a signcryption algorithm $Signcrypt$ and an unsigncryption algorithm $Unsigncrypt$, and having access to a extract oracle $O_E$, a signcryption oracle $O_S$ and an unsigncryption oracle $O_U$. $B$ first randomly chooses a number $e \leftarrow R\{1, \ldots, q\}$ and $s \leftarrow R\{1, \ldots, p\}$, where $q$ is the total number of parties that were extracted and $p$ is the total number of messages that were signcrypted throughout the run of the system. Next, $B$ runs $Z$ on the following simulated interaction with a system running $\pi_{IDSC}$ (and the dummy adversary $D$). Let $m_i$ denote the $i$-th message that $Z$ asks to signcrypt in an execution.

(1) Initially, $Z$ activates $B$. Next, $Z$ activates KGC with input $(Setup, sid)$ for $sid = (KGC, sid')$, obtains system public key $PK_s$ and hands $PK_s$ to $B$.

(2) For the first $e - 1$ times that $Z$ asks to extract some parties’ identities, $ID_i$, $B$ let-
The remaining $q - e$ times that $\mathcal{Z}$ asks to extract some party, $ID_i$, $\mathcal{B}$ lets the KGC outputs $sk_i = Extract(ID_i)$.

(3) At the $e$-th time that $\mathcal{Z}$ asks to extract a party's identity, $ID_e$, $\mathcal{B}$ queries its extract oracle $O_E$ with the pair of identities $(ID_e, ID_f)$, where $ID_f$ is the fixed identity used above, and obtains the text private key $sk_e$. It then hands $sk_e$ to $\mathcal{Z}$ as the private key of the party $P_e$ (whose identity is the $ID_e$).

(4) For the remaining $q - e$ times that $\mathcal{Z}$ asks to extract some party, $ID_i$, $\mathcal{B}$ lets the KGC outputs $sk_i = Extract(ID_i)$.

(5) For the first $s - 1$ times that $\mathcal{Z}$ asks to sign-crypt some message, $m_i$, $\mathcal{B}$ lets the sign-crypting party return $\sigma_i = Signcrypt(m_i)$.

(6) At the $s$-th time that $\mathcal{Z}$ asks to sign-crypt a message, $m_s$, $\mathcal{B}$ queries its sign-cryption oracle $O_S$ with the pair of messages $(m_s, m_f)$, where $m_f$ is the fixed message used above, and obtains the text ciphertext $\sigma_s$. It then hands $\sigma_s$ to $\mathcal{Z}$ as the sign-cryption of $m_s$.

(7) For the remaining $p - s$ times that $\mathcal{Z}$ asks to sign-crypt some message, $m_i$, $\mathcal{B}$ lets the sign-crypting party return $\sigma_i = Signcrypt(m_f)$.

(8) Whenever the unsign-crypting party $P_i$ is activated with input $(Unsigncrypt, sid, ID_s, ID_i, \sigma)$, where $ID_s$ is the identity of sign-crypting party and $\sigma = \sigma_i$ for some $i$, $\mathcal{B}$ lets $P_i$ return the corresponding plaintext $m_i$. If $\sigma$ is different from all the $\sigma_i$'s then $\mathcal{B}$ queries its unsign-cryption oracle $O_U$ on $\sigma$, obtains a value $v$, and lets $P_i$ return $v$ to $\mathcal{Z}$.

(9) When $\mathcal{Z}$ halts, $\mathcal{B}$ outputs whatever $\mathcal{Z}$ outputs and halts.

Here, assume that for some value of the security parameter $k$ we have $\text{EXEC}_{\mathcal{F}_{\text{INDSC}}, s, z}(1^k) - \text{EXEC}_{\pi_{\text{INDSC}}, d, z}(1^k) > \epsilon$. Analyzing the success probability of $B$ is done via a standard hybrid argument. Let the random variable $X_t$ denote the output of $Z$ from an interaction that is identical to an interaction with $S$ in the ideal process. It is easy to see that $X_0$ and $X_{t+1}$ are statistically close to the output of $Z$ in the ideal process, and $X_q$ and $X_{q+p}$ is identical the output of $Z$. Thus we can figure out that $B$ guesses the bit $b$ correctly in the IND-INDSC-CCA2 experiment with probability $1/2 + \epsilon(p + q)/2pq$, and the experiment $\text{EXEC}_{\mathcal{F}_{\text{INDSC}}}(1^k)$ return 1 with probability $1/2 + \epsilon/2q$. This shows that if INDSC is secure with respect to both IND-INDSC-CCA2 and EXT-INDSC-CMA then $\pi_{\text{INDSC}}$ securely realizes $\mathcal{F}_{\text{INDSC}}$.

5 Securely realizing $\mathcal{F}_{\text{SAGCOM}}$

In this section, we propose a universally composable secure authenticated group communication scheme $\pi_{\text{SAGCOM}}$ and prove that it realizes the ideal functional $\mathcal{F}_{\text{SAGCOM}}$ in $(\mathcal{F}_{\text{INDSC}}, \mathcal{F}_{\text{GKD}})$-hybrid model.

5.1 System description

We consider the network environment where there are a centralized key server and multiple multicast controllers. A sender of a group $G = \{G_1, \cdots, G_m\}$ is in charge of managing the group as a multicast controller. Let $R = \{R_1, \cdots, R_n\}$ be the universe of users. The key server as KGC generates public parameters for the system and the keys for the network group senders and users.

5.2 Protocol, $\pi_{\text{SAGCOM}}$

We propose our group communication protocol $\pi_{\text{SAGCOM}}$ in the $(\mathcal{F}_{\text{INDSC}}, \mathcal{F}_{\text{GKD}})$-hybrid model. The detailed description of $\pi_{\text{SAGCOM}}$ is shown in Figure 5.

5.3 Security proof of $\pi_{\text{SAGCOM}}$

Theorem 3. Protocol $\pi_{\text{SAGCOM}}$ securely realizes the ideal functionality $\mathcal{F}_{\text{SAGCOM}}$ in the $(\mathcal{F}_{\text{INDSC}}, \mathcal{F}_{\text{GKD}})$-hybrid model.

Proof. Let $A$ be an adversary that interacts with the parties running the protocol $\pi_{\text{SAGCOM}}$. We construct an ideal adversary $S$ such that any environment $Z$ cannot distinguish with a non-negligible probability whether it is interacting $A$ and $\pi_{\text{SAGCOM}}$ in the $(\mathcal{F}_{\text{INDSC}}, \mathcal{F}_{\text{GKD}})$-hybrid model (denoted REAL) or it is interacting with $S$ and $\mathcal{F}_{\text{SAGCOM}}$ in the ideal world (denoted IDEAL).

Construction of the ideal adversary $S$.

The adversary $S$ shown below runs a simulated copy of the adversary $A$, thus $S$ is often called a simulator. Any input from $Z$ is forwarded to $A$ and any output of $A$ is copied to the output of $S$.

1. Simulating the KGC. When an uncorrupted KGC is activated with input $(\text{SAGCOM}.\text{Setup}, \text{sid})$, $S$ obtains this value from $\mathcal{F}_{\text{SAGCOM}}$ and simulates for $A$ the protocol $\pi_{\text{SAGCOM}}$.

(1) Whenever $S$ obtains $(\text{INDSC}.\text{Setup}, \text{sid})$ from $\mathcal{F}_{\text{INDSC}}$, $S$ sends the message $(\text{INDSC}.\text{Setup}, \text{sid})$ to $A$, then forwards the response from $A$ to $\mathcal{F}_{\text{INDSC}}$. 

Protocol $\pi_{SAGCOM}$

$\pi_{SAGCOM}$ proceeds as follows, with a group of senders $G = \{G_1, \cdots, G_m\}$, a group of recipients $R = \{R_1, \cdots, R_n\}$ and $KGC$.

SAGCOM.Setup

Upon the first input $(\text{SAGCOM.Setup}, \text{sid})$, $KGC$ runs the algorithms Setup:
- The KGC obtains two key pairs $(s_G, P_G)$ and $(s_R, P_R)$, and
- outputs $(\text{SAGCOM.Setup}, \text{sid}, (s_G, P_G), (s_R, P_R))$.

SAGCOM.Extract

Upon the first input $(\text{SAGCOM.Extract}, \text{sid}, \text{ID})$, $G_i \in G$ (resp. $R_i \in R$) runs the algorithms Extract:
- The KGC sends $(\text{GKD.Distribute}, \text{sid}, s_G)$ (resp. $(\text{GKD.Distribute}, \text{sid}, s_R)$) to $\mathcal{F}_{\text{GKD}}$,
- $P_i$ (resp. $R_i$) obtains his private key $s_{G_i}$ (resp. $s_{R_i}$) from $\mathcal{F}_{\text{GKD}}$, and
- outputs $(\text{SAGCOM.Extract}, \text{sid}, (s_{G_i}, \text{ID}_{G_i}))$ (resp. $(\text{SAGCOM.Extract}, \text{sid}, (s_{R_i}, \text{ID}_{R_i}))$).

SAGCOM.Signcrypt

Upon input $(\text{SAGCOM.Signcrypt}, \text{sid}, P_R, \text{ID}_R, m)$, $G_i \in G$ runs the algorithms Signcrypt:
- $G_i \in G$ obtains $\sigma_i = \text{Signcrypt}(s_{G_i}, P_R, \text{ID}_R, m)$,
- $G_i \in G$ sends $\sigma_i$ to $R_i$, and
- outputs $(\text{SAGCOM.Signcrypt.Ciphertext}, \text{sid}, \text{ID}_R, m, \sigma_i)$.

SAGCOM.Unsigncrypt

Upon input $(\text{SAGCOM.Unsigncrypt}, \text{sid}, P_G, \text{ID}_G, \sigma)$, $R_i \in R$ runs the algorithms Unsigncrypt:
- Group signcryption generating phase
  1. $R_i$ sends $(\text{GKD.Recover}, \text{sid})$ to $\mathcal{F}_{\text{GKD}}$,
  2. $R_i$ obtains $\sigma$, and
  3. outputs $(\text{GKD.recover}, \text{sid}, \sigma)$.
- Group signcryption verifying phase
  1. $R_i$ sends $(\text{GKD.Recover}, \text{sid})$ to $\mathcal{F}_{\text{GKD}}$,
  2. $R_i$ obtains $m/\bot = \text{Unsigncrypt}(\text{ID}_G, P_G, s_{R_i}, \sigma)$ with $T \in \mathcal{T}$, and
  3. outputs $(\text{SAGCOM.Unsigncrypt.Plaintext}, \text{sid}, m/\bot, \sigma)$.

Fig. 5. The universally composable group communication protocol, $\pi_{SAGCOM}$
Whenever \( S \) obtains \((IDSC.Extract, sid, ID)\) from \( F_{IDSC} \), \( S \) sends the message \((IDSC.Extract, sid, ID)\) to \( A \), then forwards the response from \( A \) to \( F_{IDSC} \).

(3) Whenever \( S \) receives \((GKD.Distribute, sid, |s|)\) from \( F_{GKD} \), \( S \) sends the message \((GKD.Distribute, sid, |s|)\) to \( A \), then forwards the response from \( A \) to \( F_{GKD} \).

2. Simulating the sender. When an uncorrupted party \( S_i \) is activated with input \((Send, sid, T, m)\), where \( S_i \in B \subset S \) and \( T \subset R \), \( S \) obtains this value from \( F_{SAGCOM} \) and simulates for \( A \) the protocol \( \pi_{SAGCOM} \).

(1) Whenever \( S \) obtains \((Send, sid, T, l(m))\) from \( F_{SAGCOM} \), \( S \) sends the message \((Send, sid, T, l(m))\) to \( A \), then forwards the response from \( A \) to \( F_{SAGCOM} \).

(2) Whenever \( S \) receives \((IDSC.Extract, sid, PK'_i)\) from \( F_{IDSC} \), \( S \) sends the message \((IDSC.Extract, sid, PK'_i)\) to \( A \), then forwards the response \((IDSC.Extract, Ciphertext, sid, ID_i, ID, PK, \sigma)\) from \( A \) to \( F_{IDSC} \).

3. Simulating the receiver. When an uncorrupted party \( R_i \) is activated with input \((Receive, sid, B, m)\), where \( R_i \in T \subset R \) and \( B \subset S \), \( S \) obtains this value from \( F_{SAGCOM} \) and simulates for \( A \) the protocol \( \pi_{SAGCOM} \).

(1) Whenever \( S \) receives \((IDSC.Extract, sid, PK'_i)\) from \( F_{IDSC} \), \( S \) sends the message \((IDSC.Extract, sid, PK'_i)\) to \( A \), then forwards the response \((IDSC.Received, sid)\) from \( A \) to \( F_{IDSC} \).

(2) Whenever \( S \) receives \((IDSC.Unigncrypt, sid, ID_i, ID_j, PK_s, \sigma)\) from \( F_{IDSC} \), \( S \) sends the message \((IDSC.Unigncrypt, sid, ID_i, ID_j, PK_s, \sigma)\) to \( A \), then forwards the response \((IDSC.Unigncrypt, Plaintext, sid, ID_i, ID_j, PK_s, m', \sigma)\) from \( A \) to \( F_{IDSC} \).

(3) Whenever \( S \) obtains \((Receive, sid, B, l(m))\) from \( F_{SAGCOM} \), \( S \) sends the message \((Receive, sid, B, l(m))\) to \( A \), then forwards the response \((Received, sid, ok)\) from \( A \) to \( F_{SAGCOM} \).

3. Simulating party corruption. Whenever \( A \) corrupts a party, \( S \) corrupts the same party and provides \( A \) with the internal state of the corrupted party. This poses on problem since none of the parties maintains any secret information.

**IDEAL and REAL are indistinguishable.** Based on our construction of \( S \), we define three events and show that \( REAL \) and \( IDEAL \) are indistinguishable no matter which one of the three events happens.

Event 1: When a party \( P_i \in S \cup R \) is corrupted, it is easily observable that \( S \) can perfectly simulate the operations of protocols in \( REAL \) according to the logics of \( F_{IDSC}, F_{GKD} \) and \( F_{SAGCOM} \). Thus, in this case, \( REAL \) and \( IDEAL \) are indistinguishable.

Event 2: When some parties \( C \subset S \cup R \) are corrupted, if \( C \in T \) then the adversary learns the plaintext \( m \) of the ciphertext \( \sigma \); else the adversary can not learn the plaintext of the ciphertext \( \sigma \). According to \( F_{GKD} \), obviously, \( S \) can perfectly simulate the operations of protocols in \( REAL \). Therefore, \( REAL \) and \( IDEAL \) are indistinguishable.

Event 3: The receiver \( R_i \) obtains \((Received, sid, ok)\) from \( F_{SAGCOM} \) for an incoming message \((Send, sid, T, m)\), where \( R_i \in T \), while party \( S_i \) is uncorrupted at the time when the message is delivered, and has never sent \((Send, sid, T, m)\). However, according to the protocol and the logics of \( F_{IDSC}, F_{GKD} \) and \( F_{SAGCOM} \). The reason being is that, firstly the receiver should obtain a valid signcrypted ciphertext from \( F_{IDSC} \) since the protocol \( IDSC \) is IND-IDSC-CCA2 and EXT-IDSC-CMA; secondly, if an uncorrupted \( S_i \) never sent \((Send, sid, T, m)\), then the message \( m \) is never signcrypted by \( F_{IDSC} \). Thus, \( R_i \) would obtain \((Received, sid, ok)\) from \( F_{SAGCOM} \).

The above analysis shows that \( \pi_{SAGCOM} \) securely realizes the functionality \( F_{SAGCOM} \) in the \((F_{IDSC}, F_{GKD})\)-hybrid model.

6 Conclusion

We have proposed the first UC frame for secure authenticated group communication model. We have designed the identity-based signcryption functionality \( F_{IDSC} \), the group key distribution functional \( F_{GKD} \) and group communication functionality \( F_{SAGCOM} \) and subsequently, proposed a UC-secure group communication scheme that realizes \( F_{SAGCOM} \) under the \((F_{IDSC}, F_{GKD})\)-hybrid model. We have also analyzed the security of protocol \( \pi_{IDSC} \) under UC framework, which an identity-based signcryption
\( \pi_{IDSC} \) to realize the ideal functionality \( F_{IDSC} \) is proved equals to the IDSC being both IND-IDSC-CCA2 and EXT-IDSC-CMA.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 61170280, 61272398, 61262073, and 61303068; China Post-doctoral Science Foundation under Grant No. 2013M530705; The Natural Science Foundation of Guizhou of China under Grant No. 20132112; the Doctor Foundation of the Guizhou University under Grant No. 2012-024.

References