Strongly Secure One-Round Group Authenticated Key Exchange in the Standard Model

Yong Li, Zheng Yang

Ruhr-University Bochum

CANS 2013

Outline

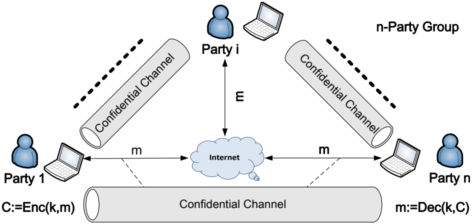
- Introduction, Motivation and Contributions
- GAKE security model (G-eCK)
- Formal definition of GAKE
- New one-round GAKE protocols in the standard model

Introduction

- Numerous group-oriented scenarios:
  - video conferencing
  - collaborative applications, etc.
- Security Goals:
  - Confidentiality
  - Integrity
  - Authentication

Introduction

- Group authenticated key exchange:
  - a shared symmetric session key for group members
  - secure multicasting network layer among the parties using a symmetric encryption with a shared session key



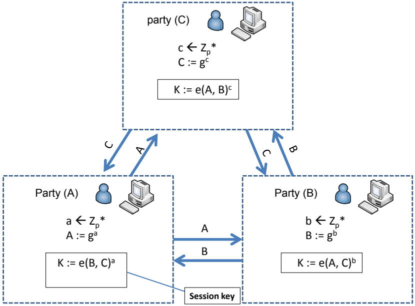
Classical example: Tripartite DHKE

- KE: Pairing-based Tripartite Diffie-Hellman key exchange (TDHKE) [AJ04]
  - Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p, generator g for $\mathbb{G}$, and a bilinear computable pairing e: $\mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_T$.
  - Party A: sk_A: $a \stackrel{\$}{\leftarrow} \mathbb{Z}_p$; $pk_A : A = g^a \in \mathbb{G}$.
  - Party B: sk_B: $b \stackrel{\$}{\leftarrow} \mathbb{Z}_p$; $pk_B : B = g^b \in \mathbb{G}$.
  - Party C: sk_C: $c \stackrel{\$}{\leftarrow} \mathbb{Z}_p$; $pk_C : C = g^c \in \mathbb{G}$.

Tripartite Diffie-Hellman Key Exchange

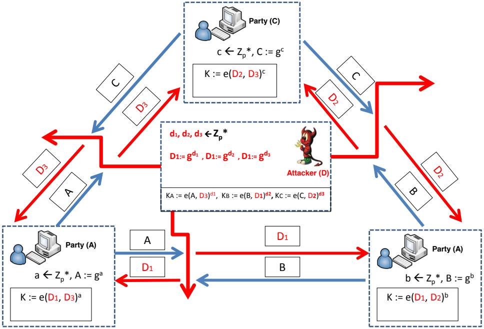
- Shared Session Key:

$$K_{A,B,C} = e(B,C)^a = e(A,C)^b = e(A,B)^c = e(g,g)^{abc}$$



Insecurity of TDHKE

- Man-in-the-Middle attack on TDHKE



How to thwart **MITM** attacks? **Authenticated Key Exchange**.

Motivation

- GAKE is a fundamental cryptographic primitive, and there are different possible security models and schemes for GAKE, e.g. [BCPQ01] [BCP02] [KY03] [BMS07], etc..
- But no secure scheme in the G-eCK security model - one of the strongest security model for one-round GAKE - under standard assumptions without random oracles.

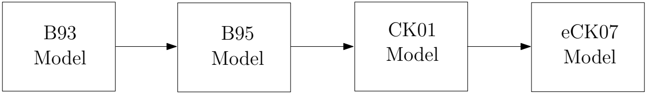
Motivation

- 2009: [MSU09] provides a tripartite/group key exchange scheme and analyses their scheme in G-eCK Security model, but with the random oracle model.
- 2012: [FMSB12] provides a tripartite key exchange. It satisfies G-eCK Security, but under the gap Bilinear Diffie-Hellman (GBDH) assumption in the random oracle model.

Contributions

- we provide a concrete construction for one-round 3AKE protocol that is G-eCK secure in the standard model - based on pairings [BS02].
- a provably G-eCK secure GAKE scheme with constant maximum group size in the standard model - based on multilinear maps [GGH13].

Evolution of AKE Security Models



B93 Model

1: Chosenn Message
2: Known Session Key

B95 Model

1: Chosenn Message
2: Known Session Key
3: Adaptive Corruption

CK01 Model

1: Chosenn Message
2: Known Session Key
3: Adaptive Corruption
3.1: Perfect Forward Secrecy
4: Leakage of Session States

eCK07 Model

1: Chosenn Message
2: Known Session Key
3: Adaptive Corruption
3.1: Weak Perfect Forward Secrecy
3.2: Key Compromise Impersonation
4: Leakage of Session States
5: Chosen Identity and Public Key

G-eCK Model: Execution Environment (1)

- a set of honest parties $\{\mathsf{ID}_1, \ldots, \mathsf{ID}_\ell\}$ for $\ell \in \mathbb{N}$ and $\mathsf{ID}_i \in \mathcal{IDS}$
- each identity is associated with a long-term key pair $(sk_{\mathsf{ID}_i}, pk_{\mathsf{ID}_i}) \in (\mathcal{SK}, \mathcal{PK})$
- each honest party ID_i can sequentially and concurrently execute the protocol multiple times with different indented partners, this is characterized by a collection of oracles $\{\pi_i^s : i \in [\ell], s \in [\rho]\}$ for $\rho \in \mathbb{N}$, i.e. Oracle π_i^s behaves as party ID_i.

G-eCK Model: Execution Environment (2)

We assume each oracle π_i^s maintains a list of independent internal state variables with following semantics:

- pid_i^s: A variable stores a set of partner identities in the group
- Φ_i^s: A variable stores the oracle decision $\Phi_i^s \in \{\mathtt{accept}, \mathtt{reject}\}$
- K_i^s: A variable records the session key $K_i^s \in \mathcal{K}_{\mathsf{KE}}$ for symmetric encryption

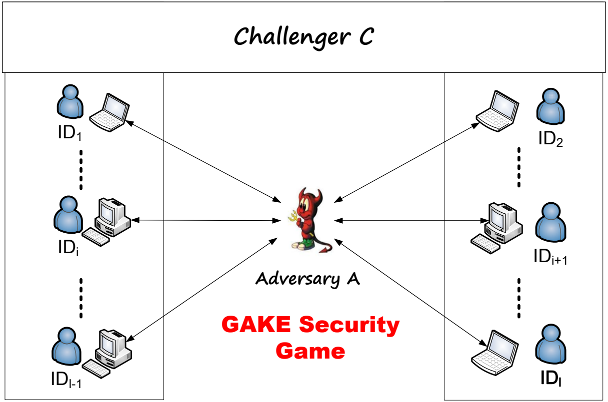
G-eCK Model: Execution Environment (2)

- st_i^s: A variable stores the maximum secret session states that are allowed to be leaked
- T_i^s: A variable stores the transcript of all messages sent and received by π_i^s during its execution

G-eCK Model: Adversarial Model (1)

Queries:

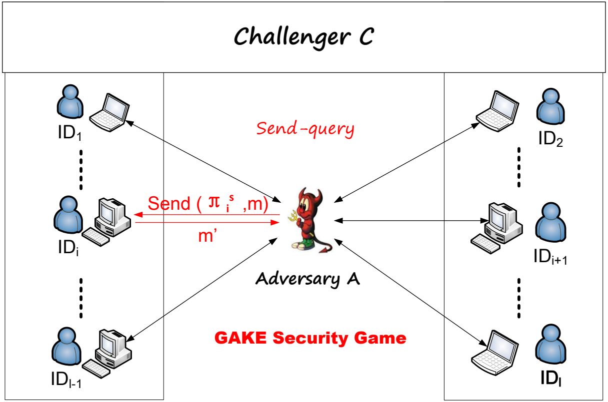
- Send
- RegisterCorrupt
- Corrupt
- RevealKey
- StateReveal
- Test



G-eCK Model: Adversarial Model (2)

Queries:

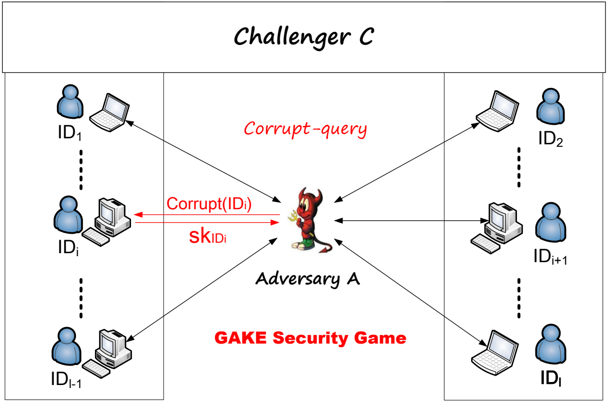
- **Send**
- RegisterCorrupt
- Corrupt
- RevealKey
- StateReveal
- Test



G-eCK Model: Adversarial Model (3)

Queries:

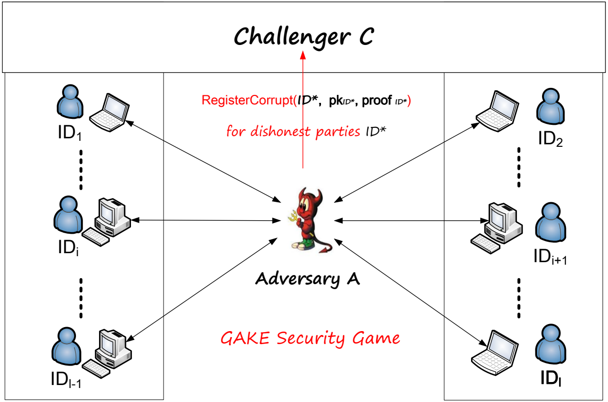
- Send
- **Corrupt**
- RegisterCorrupt
- RevealKey
- StateReveal
- Test



G-eCK Model: Adversarial Model (4)

Queries:

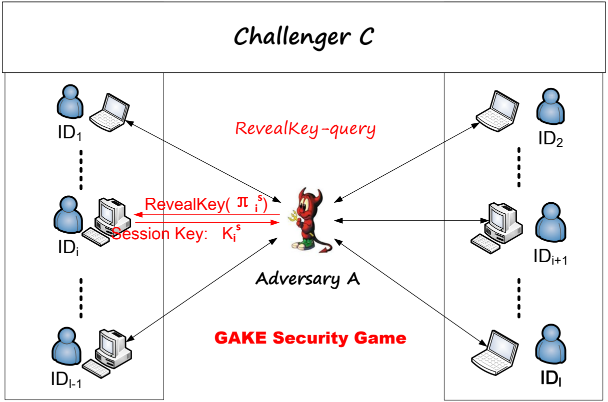
- Send
- Corrupt
- **RegisterCorrupt**
- RevealKey
- StateReveal
- Test



G-eCK Model: Adversarial Model (5)

Queries:

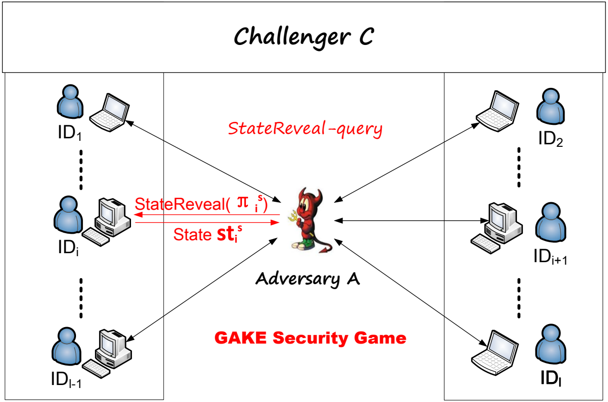
- Send
- Corrupt
- RegisterCorrupt
- **RevealKey**
- StateReveal
- Test



G-eCK Model: Adversarial Model (6)

Queries:

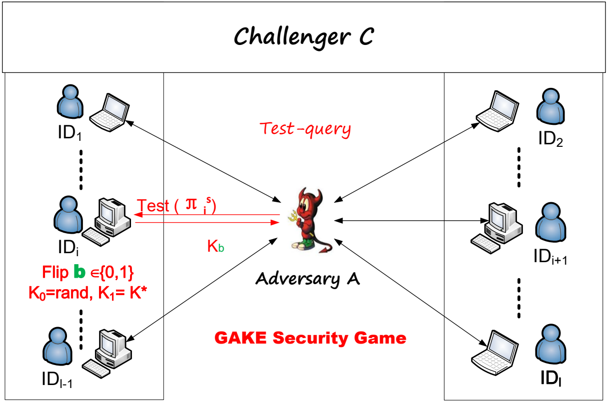
- Send
- Corrupt
- RegisterCorrupt
- RevealKey
- **StateReveal**
- Test



G-eCK Model: Adversarial Model (7)

Queries:

- Send
- Corrupt
- RegisterCorrupt
- RevealKey
- StateReveal
- **Test**



G-eCK Model: Security Game

1. Challenger $\mathcal{C}$ implements the collection of oracles $\{\pi_i^s : i \in [\ell], s \in [\rho]\}$, and generates ℓ long-term key pairs $(pk_{\mathsf{ID}_i}, sk_{\mathsf{ID}_i})$ and corresponding proof pf_i for all honest parties ID_i.
2. Adversary $\mathcal{A}$ may issue polynomial number of queries as aforementioned: Send, StateReveal, Corrupt, RegisterCorrupt and RevealKey
3. At some point, $\mathcal{A}$ may issue a Test(π_i^s) query on an oracle π_i^s during the experiment with only once.
4. At the end of the game, the $\mathcal{A}$ may terminate with outputting a bit b' as its guess for b of Test query.

G-eCK Model: Matching Sessions

We define the partnership via **matching sessions.**

Let π_i^s and π_j^t be two oracles. We say that an oracle π_i^s has a **matching session** to oracle π_j^t, if

1. $\mathsf{pid}_i^s = \mathsf{pid}_j^t$
2. π_i^s has sent all protocol messages and $\mathsf{T}_i^s = \mathsf{T}_j^t$

G-eCK Model: Freshness (1)

Let π_i^s be an accepted oracle. Let $\pi_S = \{\pi_j^t\}_{\mathsf{ID}_j \in \mathsf{pid}_i^s, j \neq i}$ be a set of oracles (if they exist), such that π_i^s has a matching session to π_j^t. The oracle π_i^s is said to be fresh if none of the following conditions holds:

- $\mathcal{A}$ queried RegisterCorrupt$(\mathsf{ID}_j, pk_{\mathsf{ID}_j}, \mathsf{pf}_{\mathsf{ID}_j})$ with some $\mathsf{ID}_j \in \mathsf{pid}_i^s$.
- $\mathcal{A}$ queried either RevealKey(π_i^s) or RevealKey(π_j^t) for some oracle $\pi_j^t \in \pi_S$.

G-eCK Model: Freshness (2)

- $\mathcal{A}$ queried both Corrupt(ID_i) and StateReveal(π_i^s).
- For some oracle $\pi_j^t \in \pi_S$, $\mathcal{A}$ queried both Corrupt(ID_j) and StateReveal(π_j^t).
- If $\mathrm{ID}_j \in \mathrm{pid}_i^s$ ($j \neq i$) and there is no oracle π_j^t such that π_i^s has a matching session to π_j^t, $\mathcal{A}$ queried Corrupt(ID_j).

G-eCK Model: Security Definition

We say that an adversary $\mathcal{A}$ (t, ϵ)-breaks the G-eCK security of a correct group AKE protocol Σ, if $\mathcal{A}$ runs the security game within time t, and the following condition holds:

- If a Test query has been issued to an oracle π_i^s without failure and π_i^s is fresh throughout the security game, then the probability that the bit b' returned by $\mathcal{A}$ equals to the bit b chosen by the Test query is bounded by

$$|\Pr[b = b'] - 1/2| > \epsilon,$$

We say that a correct group AKE protocol Σ is (t, ϵ)-g-eCK-secure, if there exists no adversary that (t, ϵ)-breaks the g-eCK security of Σ.

Formal Definition of One-round GAKE (1)

We consider the following variables:

- $\mathcal{PK}$: a longterm key space for public key and private key
- $\mathcal{SK}$: a longterm key space for private key
- $\mathcal{R}_{\mathsf{ORGAKE}}$: a randomness space
- $\mathcal{IDS}$: an identity space
- $\mathcal{K}_{\mathsf{ORGAKE}}$: a shared session key space
- $\mathsf{GD} := ((\mathsf{ID}_1, pk_{\mathsf{ID}_1}), \ldots, (\mathsf{ID}_n, pk_{\mathsf{ID}_n}))$: a list which is used to store the public information of a group of parties
- T: the transcript storing the messages sent and received by a protocol instance at a party which are sorted orderly.

Formal Definition of One-round GAKE (2)

A ORGAKE scheme consists of 4 algorithms:

- $pms \leftarrow \mathrm{Setup}(1^\kappa)$
  - output: a set of system parameters storing in a variable *pms*.
- $(sk_{\mathrm{ID}}, pk_{\mathrm{ID}}, \mathrm{pf}_{\mathrm{ID}}) \xleftarrow{\$} \mathrm{ORGAKE.KGen}(pms, \mathrm{ID})$
  - output: $(sk_{\mathrm{ID}}, pk_{\mathrm{ID}}) \in \{\mathcal{PK}, \mathcal{SK}\}$ for party ID and a non-interactive proof $\mathrm{pf}_{\mathrm{ID}}$ for pk_{ID} which is required during key registration.
- $m_{\mathrm{ID}_i} \xleftarrow{\$} \mathrm{ORGAKE.MF}(pms, sk_{\mathrm{ID}_i}, r_{\mathrm{ID}_i}, \mathrm{GD})$
  - output: a message m_{ID_i} to be sent in a protocol pass.
- $K \leftarrow \mathrm{ORGAKE.SKG}(pms, sk_{\mathrm{ID}_i}, r_{\mathrm{ID}_i}, \mathrm{GD}, \mathrm{T})$
  - output: session key $K \in \mathcal{K}_{\mathrm{ORGAKE}}$.

Formal Definition of One-round GAKE (3): Correctness

For correctness, on input the same transcript T and group description GD = $((\mathrm{ID}_1, pk_1), \ldots, (\mathrm{ID}_n, pk_n))$, algorithm ORGAKE.SKG satisfies the constraint:

- ORGAKE.SKG $(pms, sk_{\mathrm{ID}_1}, r_{\mathrm{ID}_1}, \mathrm{GD}, \mathrm{T})$ = ORGAKE.SKG $(pms, sk_{\mathrm{ID}_i}, r_{\mathrm{ID}_i}, \mathrm{GD}, \mathrm{T})$,

Stongly Secure One-Round GAKE Schemes in the Standard Model

Building blocks:

- A Target Collision Resistant Hash Function (TCRHF)
- A Pseudo-Random Function (PRF)
- A Weak Programmable Hash Function wPHF [HJK11]

TAKE in the Standard Model from Bilinear Maps

Tripartite AKE Protocol Execution:

- **Setup:**
  - Symmetric bilinear groups $\mathcal{PG} = (\mathbb{G}, g, \mathbb{G}_T, p, e) \stackrel{\$}{\leftarrow} \mathsf{PG.Gen}(1^\kappa)$ and a set of random values $\{u_i\}_{1 \leq i \leq 4} \stackrel{\$}{\leftarrow} \mathbb{G}$
  - A target collision resistant hash function $\mathsf{TCRHF}(hk_{\mathsf{TCRHF}}, \cdot) : \mathcal{K}_{\mathsf{TCRHF}} \times \mathbb{G} \rightarrow \mathbb{Z}_p$, where $hk_{\mathsf{TCRHF}} \stackrel{\$}{\leftarrow} \mathsf{TCRHF.KGen}(1^\kappa)$
  - A pseudo-random function family $\mathsf{PRF}(\cdot, \cdot) : \mathbb{G}_T \times \{0,1\}^* \rightarrow \mathcal{K}_{\mathsf{AKE}}$.

TAKE in the Standard Model from Bilinear Maps

- **Long-term Key Generation and Registration**: Input $pms := (\mathcal{PG}, \{u_i\}_{1 \leq i \leq 4}, hk_{\text{TCRHF}})$, each party runs as:
  - **Party $\hat{A}$**:
    $sk_{\hat{A}} = a \xleftarrow{\$} \mathbb{Z}_p^*$, $h_A = \text{TCRHF}(A)$
    $pk_{\hat{A}} = (A, t_A) = (A = g^a, t_A = (u_4^{h_A^3} u_3^{h_A^2} u_2^{h_A} u_1)^a)$
  - **Party $\hat{B}$**:
    $sk_{\hat{B}} = b \xleftarrow{\$} \mathbb{Z}_p^*$, $h_B = \text{TCRHF}(B)$
    $pk_{\hat{B}} = (B, t_B) = (A = g^b, t_B = (u_4^{h_B^3} u_3^{h_B^2} u_2^{h_B} u_1)^b)$
  - **Party $\hat{C}$**:
    $sk_{\hat{C}} = c \xleftarrow{\$} \mathbb{Z}_p^*$, $h_C = \text{TCRHF}(C)$
    $pk_{\hat{C}} = (C, t_C) = (C = g^c, t_C = (u_4^{h_C^3} u_3^{h_C^2} u_2^{h_C} u_1)^c)$

TAKE in the Standard Model from Bilinear Maps

- **Ephemeral Key Generation and Broadcast Messages**:
  - **Party** $\hat{A}$: $x \xleftarrow{\$} \mathbb{Z}_p^*$, $X := g^x$
    $h_X :=$ TCRHF(X),$t_X := (u_4^{h_X^3} u_3^{h_X^2} u_2^{h_X} u_1)^x$
    $\hat{A}$ broadcasts messages $(\hat{A}, A, t_A, X, t_X)$ to $\hat{B}$ and $\hat{C}$.
  - **Party** $\hat{B}$: $y \xleftarrow{\$} \mathbb{Z}_p^*$, $Y := g^y$
    $h_Y :=$ TCRHF(Y), $t_Y := (u_0 u_1^{h_Y} u_2^{h_Y^2} u_3^{h_Y^3})^y$
    $\hat{B}$ broadcasts messages $(\hat{B}, B, t_B, Y, t_Y)$ to $\hat{A}$ and $\hat{C}$.
  - **Party** $\hat{C}$: $z \xleftarrow{\$} \mathbb{Z}_p^*$, $Z := g^z$
    $h_Z :=$ TCRHF(Z), $t_Z := (u_0 u_1^{h_Z} u_2^{h_Z^2} u_3^{h_Z^3})^z$
    $\hat{A}$ broadcasts messages $(\hat{C}, C, t_C, Z, t_Z)$ to $\hat{A}$ and $\hat{B}$.

TAKE in the Standard Model from Bilinear Maps

- **Session Key Generation (1):**
  Upon receiving $(\hat{B}, B, t_B, Y, t_Y)$ and $(\hat{C}, C, t_C, Z, t_Z)$, **party** $\hat{A}$ computes the session key as follows:
  - sid $:= \hat{A}||A||t_A||X||t_X||\hat{B}||B||t_B||Y||t_Y||\hat{C}||C||t_C||Z||t_Z$
  - $h_B = \text{TCRHF}(B)$, $h_C = \text{TCRHF}(C)$, $h_Y = \text{TCRHF}(Y)$ and $h_Z = \text{TCRHF}(Z)$

TAKE in the Standard Model from Bilinear Maps

- **Session Key Generation (2):**
  - if $e(t_B, g) \neq e(u_0 u_1^{h_B} u_2^{h_B^2} u_3^{h_B^3}, B)$ or
    $e(t_C, g) \neq e(u_0 u_1^{h_C} u_2^{h_C^2} u_3^{h_C^3}, C)$ or
    $e(t_Y, g) \neq e(u_0 u_1^{h_Y} u_2^{h_Y^2} u_3^{h_Y^3}, Y)$ or
    $e(t_Z, g) \neq e(u_0 u_1^{h_Z} u_2^{h_Z^2} u_3^{h_Z^3}, Z)$
    - then “rejects”
    - else $k := e(BY, CZ)^{a+x}$ and $k_e := \mathrm{PRF}(k, sid)$
  - Return the session key: k_e

TAKE in the Standard Model from Bilinear Maps

- Upon receiving $(\hat{A}, A, t_A, X, t_X)$ and $(\hat{C}, C, t_C, Z, t_Z)$, **party** $\hat{B}$ proceeds as the same as **party** $\hat{A}$:
  $k := e(AX, CZ)^{b+z}$ and $k_e := \mathrm{PRF}(k, sid)$
- Upon receiving $(\hat{A}, A, t_A, X, t_X)$ and $(\hat{B}, B, t_B, Y, t_Y)$, party $\hat{C}$ proceeds as the same as **party** $\hat{A}$:
  $k := e(AX, BY)^{a+x}$ and $k_e := \mathrm{PRF}(k, sid)$

TAKE in the Standard Model from Bilinear Maps

$\hat{A}$	$\hat{B}$	$\hat{C}$
$sk_{\hat{A}} = a \xleftarrow{\$} \mathbb{Z}_p^*$	$sk_{\hat{B}} = b \xleftarrow{\$} \mathbb{Z}_p^*$	$sk_{\hat{C}} = c \xleftarrow{\$} \mathbb{Z}_p^*$
$pk_{\hat{A}} = (A, t_A) := (g^a, (u_4^{h_A^3} u_3^{h_A^2} u_2^{h_A} u_1)^a)$	$pk_B = (B, t_B) := (g^b, (u_4^{h_B^3} u_3^{h_B^2} u_2^{h_B} u_1)^b)$	$pk_{\hat{C}} = (C, t_C) := (g^c, (u_4^{h_C^3} u_3^{h_C^2} u_2^{h_C} u_1)^c)$
$x \xleftarrow{\$} \mathbb{Z}_p^*,\ X = g^x$	$y \xleftarrow{\$} \mathbb{Z}_p^*,\ Y = g^y$	$z \xleftarrow{\$} \mathbb{Z}_p^*,\ Z = g^z$
$t_X := (u_4^{h_X^3} u_3^{h_X^2} u_2^{h_X} u_1)^x$	$t_Y := (u_4^{h_Y^3} u_3^{h_Y^2} u_2^{h_Y} u_1)^y$	$t_Z := (u_4^{h_Z^3} u_3^{h_Z^2} u_2^{h_Z} u_1)^z$
$\mathbf{broadcast}(\hat{A}, A, t_A, X, t_X)$	$\mathbf{broadcast}(\hat{B}, B, t_B, Y, t_Y)$	$\mathbf{broadcast}(\hat{C}, C, t_C, Z, t_Z)$
$h_B := TCRHF(B)$ $h_Y := TCRHF(Y)$ $h_C := TCRHF(C)$ $h_Z := TCRHF(Z)$	$h_A := TCRHF(A)$ $h_X := TCRHF(X)$ $h_C := TCRHF(C)$ $h_Z := TCRHF(Z)$	$h_A := TCRHF(A)$ $h_X := TCRHF(X)$ $h_B := TCRHF(B)$ $h_Y := TCRHF(Y)$
$U_B := u_4^{h_B^3} u_3^{h_B^2} u_2^{h_B} u_1$ $U_Y := u_4^{h_Y^3} u_3^{h_Y^2} u_2^{h_Y} u_1$ $U_C := u_4^{h_C^3} u_3^{h_C^2} u_2^{h_C} u_1$ $U_Z := u_4^{h_Z^3} u_3^{h_Z^2} u_2^{h_Z} u_1$	$U_A := u_4^{h_A^3} u_3^{h_A^2} u_2^{h_A} u_1$ $U_X := u_4^{h_X^3} u_3^{h_X^2} u_2^{h_X} u_1$ $U_C := u_4^{h_C^3} u_3^{h_C^2} u_2^{h_C} u_1$ $U_Z := u_4^{h_Z^3} u_3^{h_Z^2} u_2^{h_Z} u_1$	$U_A := u_4^{h_A^3} u_3^{h_A^2} u_2^{h_A} u_1$ $U_X := u_4^{h_X^3} u_3^{h_X^2} u_2^{h_X} u_1$ $U_B := u_4^{h_B^3} u_3^{h_B^2} u_2^{h_B} u_1$ $U_Y := u_4^{h_Y^3} u_3^{h_Y^2} u_2^{h_Y} u_1$
Reject if either $e(t_B, g) \neq e(U_B, B)$ or $e(t_C, g) \neq e(U_C, C)$ or $e(t_Y, g) \neq e(U_Y, Y)$ or $e(t_Z, g) \neq e(U_Z, Z)$	Reject if either $e(t_A, g) \neq e(U_A, A)$ or $e(t_C, g) \neq e(U_C, C)$ or $e(t_X, g) \neq e(U_X, X)$ or $e(t_Z, g) \neq e(U_Z, Z)$	Reject if either $e(t_A, g) \neq e(U_A, A)$ or $e(t_B, g) \neq e(U_B, B)$ or $e(t_X, g) \neq e(U_X, X)$ or $e(t_Y, g) \neq e(U_Y, Y)$

$$sid = \hat{A}\|A\|t_A\|X\|t_X\|\hat{B}\|B\|t_B\|Y\|t_Y\|\hat{C}\|C\|t_C\|Z\|t_Z$$

$k := e(BY, CZ)^{a+x}$	$k := e(AX, CZ)^{b+y}$	$k := e(AX, BY)^{c+z}$
$K_e := \mathsf{PRF}(k, sid)$	$K_e := \mathsf{PRF}(k, sid)$	$K_e := \mathsf{PRF}(k, sid)$

Target Collision Resistant Hash Function	(3, Poly)-wPHF	PRF

Security of TAKE in the Standard Model

- Cube Bilinear Decisional Diffie-Hellman Assumption (CBDDH)
  - Let $\mathcal{PG} = (\mathbb{G}, g, \mathbb{G}_T, p, e)$ denote the description of symmetric bilinear group
  - Given (g, g^a, T) decide whether or not $T = e(g, g)^{a^3}$

Security of TAKE in the Standard Model

Theorem 1:

Assume each ephemeral key chosen during key exchange has bit-size $\lambda \in \mathcal{N}$. Suppose that the CBDDH problem is $(t, \epsilon_{\mathsf{CBDDH}})$-hard in the symmetric bilinear groups $\mathcal{PG}$, the TCRHF is $(t, \epsilon_{\mathsf{TCRHF}})$-secure target collision resistant hash function family, and the PRF is $(t, \epsilon_{\mathsf{PRF}})$-secure pseudo-random function family. Then the proposed protocol is (t', ϵ)-session-key-secure with $t' \approx t$ and $\epsilon \leq \frac{(\rho\ell)^2}{2^\lambda} + \epsilon_{\mathsf{TCRHF}} + 4(\rho\ell)^3 \cdot \epsilon_{\mathsf{CBDDH}} + \epsilon_{\mathsf{PRF}}$.

GAKE in the Standard Model from Multilinear Groups

GAKE Protocol Execution:

- **Setup**:
  - n-mulitilinear groups $\mathcal{MLG} = (\mathbb{G}, \mathbb{G}_T, g, p, me) \xleftarrow{\$} \text{MLG.Gen}(\kappa, n)$, a set of random values $\{u_j\}_{0 \leq j \leq n+1} \xleftarrow{\$} \mathbb{G}$ and a random element $\Phi \xleftarrow{\$} \mathbb{G}$ denoted here as padding for achieving scalability.
  - a target collision resistant hash function $\text{TCRHF}(hk_{\text{TCRHF}}, \cdot) : \mathcal{K}_{\text{TCRHF}} \times \mathbb{G} \rightarrow \mathbb{Z}_p$, where $hk_{\text{TCRHF}} \xleftarrow{\$} \text{TCRHF.KGen}(1^\kappa)$
  - a pseudo-random function family $\text{PRF}(\cdot, \cdot) : \mathbb{G}_T \times \{0,1\}^* \rightarrow \mathcal{K}_{\text{AKE}}$

GAKE in the Standard Model from Multilinear Groups

- **Long-term Key Generation and Registration**:
  On input $pms := (\mathcal{MLG}, \{u_j\}_{0 \leq j \leq n+1}, hk_{\text{TCRHF}})$, each **Party** $\hat{D}_i$ $(2 \leq i \leq n+1)$ runs as follows:
  - **Party** $\hat{D}_i$ computes:
    $sk_{\hat{D}_i} = d_i \xleftarrow{\$} \mathbb{Z}_p^*$ and $h_{D_i} = \text{TCRHF}(D_i)$
    $pk_{\hat{D}_i} = (D_i, t_{D_i}) = (D_i = g^{d_i}, t_{D_i} = (\prod_{j=0}^{n+1} u_j^{h_{D_i}^j})^{d_i})$

GAKE in the Standard Model from Multilinear Maps

Let ω denote the size of group for a protocol instance such that $2 \leq \omega \leq n+1$.

- **Ephemeral Key Generation and Broadcast Messages:**
  - **party** $\hat{D}_i$:
    $x_i \xleftarrow{\$} \mathbb{Z}_p^*$, $X_i := g^{x_i}$
    $h_{X_i} := \mathrm{TCRHF}(X_i)$, $t_{X_i} = (\prod_{j=0}^{n+1} u_j^{h_{X_i}^j})^{x_i}$
    $\hat{D}_i$ broadcasts messages $(\hat{D}_i, D_i, t_{D_i}, X_i, t_{X_i})$ to its intended communication partners.

GAKE in the Standard Model from Multilinear Maps

- **Session Key Generation (1):**
  Upon receiving all messages $\{\hat{D}_l, D_l, t_{D_l}, X_l, t_{X_l}\}_{1 \le l \le \omega, l \ne i}$ from each session participant, **party** $\hat{D}_i$ computes the session key as follows:
  - sid $:= \hat{D}_1 || D_1 || t_{D_1} || X_1 || t_{X_1} || \ldots || \hat{D}_\omega || D_\omega || t_{D_\omega} || X_\omega || t_{X_\omega}$
  - $h_{D_l} = \mathrm{TCRHF}(D_l)$, $h_{X_l} = \mathrm{TCRHF}(X_l)$, where $1 \le l \le \omega, l \ne i$

GAKE in the Standard Model from Multilinear Maps

- **Session Key Generation (2):**
  - if $me(t_{D_I}, g, \ldots, g) \neq me(\prod_{j=0}^{n+1} u_j^{h^j_{D_I}}, D_I, g, \ldots, g)$ or $me(t_{X_I}, g, \ldots, g) \neq me(\prod_{j=0}^{n+1} u_j^{h^j_{X_I}}, X_I, g, \ldots, g)$
    - then "rejects"
    - else $k :=$ $me(D_1X_1, \ldots, D_{i-1}X_{i-1}, D_{i+1}X_{i+1}, \ldots, D_\omega X_\omega, \underbrace{\Phi, \ldots, \Phi}_{(n+1-\omega)\ \Phi})^{d_i + x_i}$

      and $k_e := \mathrm{PRF}(k, sid)$
  - Return the session key: k_e

Security of GAKE in the Standard Model

- n-Multiliear Decisional Diffie-Hellman Assumption (nMDDH)
  - Let $\mathcal{MLG} = (\mathbb{G}, \mathbb{G}_T, g, p, me)$ denote the description of n-multilinear groups
  - Given (g, g^a, T) decide whether or not $T = me(g, \ldots, g)^{a^{n+1}}$

Security of TAKE in the Standard Model

Theorem 2:

Assume each ephemeral key chosen during key exchange has bit-size $\lambda \in \mathcal{N}$. Suppose that the nMDDH problem is $(t, \epsilon_{\mathsf{nMDDH}})$-hard in the symmetric multilinear $\mathcal{MLP}$, the TCRHF is $(t, \epsilon_{\mathsf{TCRHF}})$-secure target collision resistant hash function family, and the PRF is $(t, \epsilon_{\mathsf{PRF}})$-secure pseudo-random function family. Then the proposed protocol of size $2 \leq \omega \leq n+1$ is (t', ϵ)-session-key-secure with $t' \approx t$ and $\epsilon \leq \frac{(d\ell)^{n+1}}{2^{\lambda_1}} + \epsilon_{\mathsf{TCRHF}} + (n+2)(d\ell)^{n+1} \cdot \epsilon_{\mathsf{nMDDH}} + \epsilon_{\mathsf{PRF}}$.

Thank you for your attention!

Bibliography

- [BCPQ01] Provably authenticated group Diffie-Hellman key exchange. Bresson, Chevassut, Pointcheval, and Quisquater - ACM CCS 01.
- [BCP02] Dynamic group Diffie-Hellman key exchange under standard assumptions. Bresson, Chevassut, and Pointcheval â EUROCRYPT 2002.
- [BCP02] Scalable protocols for authenticated group key exchange. Jonathan Katz and Moti Yung - CRYPTO 2003.
- [BS02] Applications of multilinear forms to cryptography. Dan Boneh and Alice Silverberg - 2002.

Bibliography

- [AJ04] A one round protocol for tripartite Diffie-Hellman. Antoine Joux - Journal of Cryptology, 2004.
- [BMS07] On security models and compilers for group key exchange protocols. Bresson, Manulis, and Schwenk - IWSEC 07.
- [HofKil08] Programmable hash functions and their applications. Hofheinz, Kiltz - CRYPTO 08
- [MSU09] Modeling leakage of ephemeral secrets in tripartite/group key exchange. Manulis, Suzuki, and Ustaoglu - ICISC 09

Bibliography

- [GBN09] Modeling key compromise impersonation attacks on group key exchange protocols. Gorantla, Boyd, and Nieto - PKC 2009.
- [HJK11] Short signatures from weaker assumptions. Hofheinz, Jager, and Kiltz â ASIACRYPT 2011.
- [FMSB12] Sufficient condition for ephemeral key-leakage resilient tripartite key exchange. Fujioka, Manulis, Suzuki, and Ustaoglu - ACISP 12.
- [GGH13] Candidate multilinear maps from ideal lattices. Garg, Gentry, and Halevi - EUROCRYPT-2013.