Group Key Exchange Enabling On-Demand Derivation of P2P Keys

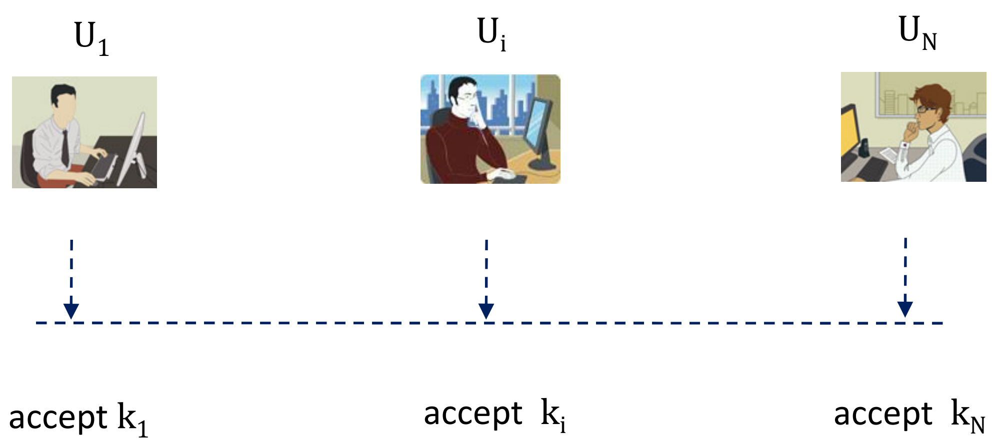
Mark Manulis
Cryptographic Protocols Group
TU Darmstadt & CASED

CRYPTOGRAPHIC PROTOCOLS

Group Key Exchange

Users in $\mathbf{U} = \{U_1, ..., U_N\}$ run a **Group Key Exchange (GKE)** Protocol and compute a session group key k indistinguishable from $k^* \in_R \{0,1\}^\kappa$

a nice building block for group applications

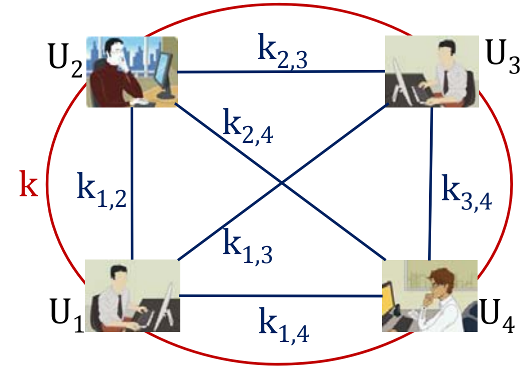


$k_1 = k_2 = ... = k_N$

secure (private and authenticated) group channel for $U_1, ..., U_N$

Main Goal: Extending GKE with P2P Keys

One protocol ⇒ **1 group key** + **up to N peer-2-peer keys.**
All keys must be <u>independent</u> (across different sessions).
Denote such protocols **GKE+P**.



Naive solutions

1. Execute GKE within $\mathbf{U}$ and own 2KE between each U_i and U_j in parallel.
   Drawback Gives all N keys at once but needs $(n^2 - n)/2$ parallel 2KE sessions.

2. Execute GKE within $\mathbf{U}$ followed by *on-demand* execution of 2KE between U_i and U_j.
   Drawback Up to $(n - 1)$ additional 2KE sessions per U_i.

Can we do better?

Since users interact in GKE can we derive p2p keys *non-interactively*?

Group Diffie-Hellman Key Exchange

Many GKE Protocols

are <u>extensions</u> of 2-party DHKE (Diffie-Hellman‘76) to a group setting

GroupDH

is a GKE protocol amongst the users in $U = \{U_1, \ldots, U_N\}$ in which each U_i chooses own exponent $x_i \in_R \mathbb{Z}_Q$ and computes $k'_i = f(g, x_1, \ldots, x_N)$ for some $f : \mathbb{G} \times \mathbb{Z}_Q^N \rightarrow \mathbb{G}$. A GroupDH protocol is *secure* if k'_i is indistinguishable from $k^* \in_R \mathbb{G}$.

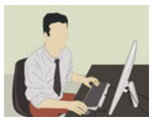
Examples

(protocols with passive security) Steer-Strawczynski-Diffie-Wiener‘88, Ingemarsson-Tang-Wong‘89, Burmester-Desmedt‘94, Steiner-Tsudik-Waidner‘96, Kim-Perrig-Tsudik’04, Nam-Paik-Kim-Won‘07, Desmedt-Lange’08

and their (authenticated) variants

Diffie-Hellman Key Exchange

Let $Q, P \in$ *PRIMES*, $Q \mid P - 1$ and $\mathbb{G} = \langle g \rangle$ a cyclic subgroup of $\mathbb{Z}^*_P$ of order Q

U_1	DHKE	U_2
		
$x_1 \in_R \mathbb{Z}^*_Q$		$x_2 \in_R \mathbb{Z}^*_Q$
$y_1 = g^{x1}$	y_1 →	$y_2 = g^{x2}$
	← y_2	
accept $k' = y_2^{x1}$		accept $k' = y_1^{x2}$

$k' = g^{x1x2}$

secure against *eavesdropping* attacks under the DDH assumption

$$Adv_{DDH}(A') = max_{A'}|Pr_{a,b}[A'(g, g^a, g^b, g^{ab}) = 1] - Pr_{a,b,c}[A'(g, g^a, g^b, g^c) = 1]| \leq \epsilon(|Q|)$$

security is defined in the sense of *indistinguishability* of k' from $k^* \in_R \mathbb{G}$

Parallel Diffie-Hellman Key Exchange

Let $\mathbf{U} = \{U_1, \ldots, U_N\}$ be a set of users (their *unique* identities).

PDHKE

U_1

$x_1 \in_R \mathbb{Z}^*_Q$

$y_1 = g^{x1}$

y_1

U_i

$x_i \in_R \mathbb{Z}^*_Q$

$y_i = g^{xi}$

y_i

U_N

$x_N \in_R \mathbb{Z}^*_Q$

$y_N = g^{xN}$

y_N

"broadcast"

asynchronous, point-to-point

accept $\{k'_{1,j} = y_j^{x1}\}_j$

accept $\{k'_{i,j} = y_j^{xi}\}_j$

accept $\{k'_{N,j} = y_j^{xN}\}_j$

U_i computes *peer-2-peer keys* $k'_{i,1} = g^{x_i x_1}, k'_{i,2} = g^{x_i x_2}, \ldots, k'_{i,N} = g^{x_i x_N}$

Passive Security Setting for PDHKE

Passive attacks (Canetti-Krawczyk'01)

more than just eavesdropp, i.e. also drop, delay, change order of messages

corrupt U and choose messages on behalf of U

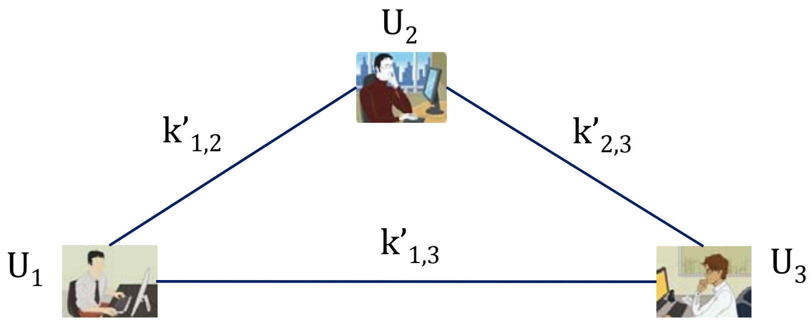
<u>but</u> *no* impersonation (via modification, injection, or replay) of uncorrupted users

Basic security goal for PDHKE

indistinguishability of a p2p key $k'_{i,j}$ accepted by U_i and U_j from $k^* \in_R \mathbb{G}$

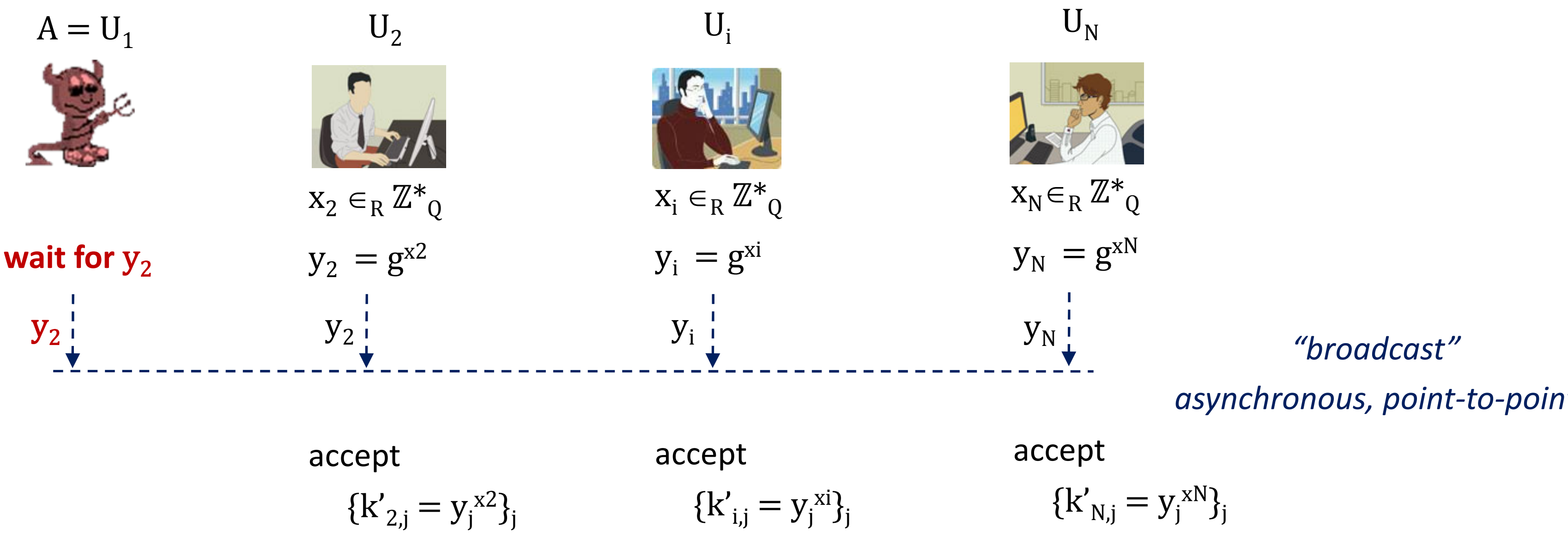
<u>U_i and U_j are uncorrupted</u> upon computation of $k'_{i,j}$ <u>but any other U can be corrupted</u>

independence of $k'_{i,j}$ from other p2p keys (also from those computed by U_i, U_j)



knowledge of $k'_{1,2}$ should *not* reveal any information about $k'_{1,3}$ and $k'_{2,3}$

Simple Attack on PDHKE



A does not know x_2

but each U_i computes $\{k'_{i,1} = g^{x_i x_2}\}_i = \{k'_{i,2} = g^{x_i x_2}\}_i$

$\Downarrow$

A can distinguish any $k'_{i,2} = g^{x_i x_2}$ from k^* by revealing $k'_{i,1}$ from U_i

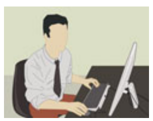
P2P Key Derivation in PDHKE

$\mathbf{U} = \{U_1, \ldots, U_N\}$. Hash function $H_p : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$. Cyclic group $\mathbb{G} = (g, P, Q)$.

For each pair (U_i, U_j) the input order to H is determined by $i < j$ (to ensure $k_{i,j} = k_{j,i}$)

PDHKE + Hash-based Key Derivation

U_i



$x_i \in_R \mathbb{Z}^*_Q$

$y_i = g^{x_i}$ — y_i ⟶

$\{k'_{i,j} = y_j^{x_i}\}_j$ ⟵ $\{y_j\}_j$

accept

$\{k_{i,j} = H_p(k'_{i,j}, (U_i, y_i), (U_j, y_j))\}_j$

$k_{i,j} = H_p(k'_{i,j}, (U_i, y_i), (U_j, y_j))$ 

uniqueness of user ids ⇒ uniqueness of hash inputs $H_p(*, (U_i, *), (U_j, *))$

for any uncorrupted U_i and at most q invoked sessions

$$\Pr[k_{i,j} \text{ occurs twice}] \leq \frac{Nq^2}{Q} + \frac{q_{H_p}^{\;2}}{2^\kappa}$$

Benefits of PDHKE

Users in $\mathbf{U} = \{U_1, \ldots, U_N\}$ run PDHKE and

obtain up to N independent peer-2-peer secure channels

investing the *optimal* amount of communication costs

1 round, 1 message per U_i (consisting of 1 element from $\mathbb{G}$)

and low computation costs

1 exponentiation and 1 hash computation per $k_{i,j}$

with possibility to compute pairwise keys *on-demand w/o further communication*

each U_i stores x_i and $\{y_j\}_j$ and can derive any $k_{i,j}$ if this becomes necessary

gives us a *compiler* from GKE to GKE+P (sequential composition of PDHKE||GKE)

also interesting as a stand-alone group application

Merge GroupDH with PDHKE

Optimization idea

Let $U_i \in \mathbf{U}$ *re-use* $x_i \in \mathbb{Z}_Q$ from GroupDH to compute the p2p key $k_{i,j}$ with $U_i \in \mathbf{U}$ (by applying the PDHKE technique).

Suitable key derivation

Hash functions $H_g, H_p : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$. Let $k'_i = f(g, x_1, ..., x_N)$.

Group key $k_i = H_g(k'_i, (U_1, y_1), ..., (U_N, y_N))$

Pairwise key $k_{i,j} = H_p(k'_{i,j}, (U_i, y_i), (U_j, y_j))$ where $k'_{i,j} = y_j^{x_i}$ (assuming $i < j$)

Suitable GroupDH protocols (protocols with passive security)

Protocols in which each U_i broadcasts $y_i = g^{x_i}$.

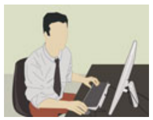
in this talk

Burmester-Desmedt'94 (2 rounds, broadcast complexity O(n))

Kim-Perrig-Tsudik'04 (2 rounds, broadcast complexity O(n), Tree-Diffie-Hellman method)

Burmester-Desmedt GroupDH Protocol

Cyclic group $\mathbb{G} = (g, P, Q)$. $U_1, \ldots, U_N$ are arranged into a *cycle* s.t. $U_0 = U_N$, $U_{N+1} = U_1$.

	U_{i-1}	U_i	U_{i+1}
			
	$x_{i-1} \in_R \mathbb{Z}_Q$	$x_i \in_R \mathbb{Z}_Q$	$x_{i+1} \in_R \mathbb{Z}_Q$
	$y_{i-1} = g^{x_{i-1}}$	$y_i = g^{x_i}$	$y_{i+1} = g^{x_{i+1}}$
Round 1	↓	↓	↓
	$z_{i-1} = (y_i/y_{i-2})^{x_{i-1}}$	$z_i = (y_{i+1}/y_{i-1})^{x_i}$	$z_{i+1} = (y_{i+2}/y_i)^{x_{i+1}}$
Round 2	↓	↓	↓

Group DH element $k'_i = y_{i-1}^{Nx_i} z_i^{N-1} z_{i+1}^{N-2} \ldots z_{i+N-2} = g^{x_1x_2 + x_2x_3 + \ldots + x_{N-1}x_N}$

Group key $k_i = H_g(g^{x_1x_2 + x_2x_3 + \ldots + x_{N-1}x_N}, (U_1, y_1), \ldots, (U_N, y_N))$

Pairwise key $k_{i,j} = H_p(g^{x_ix_j}, (U_i, y_i), (U_j, y_j))$

Is this secure?

Analysis of PDHKE-BD

Group key $k_i = H_g(g^{x_1x_2 + x_2x_3 + \ldots + x_{N-1}x_N}, (U_1, y_1), \ldots, (U_N, y_N))$

Pairwise key $k_{i,j} = H_p(g^{x_ix_j}, (U_i, y_i), (U_j, y_j))$

Is this secure?
No.

Individual Attacks

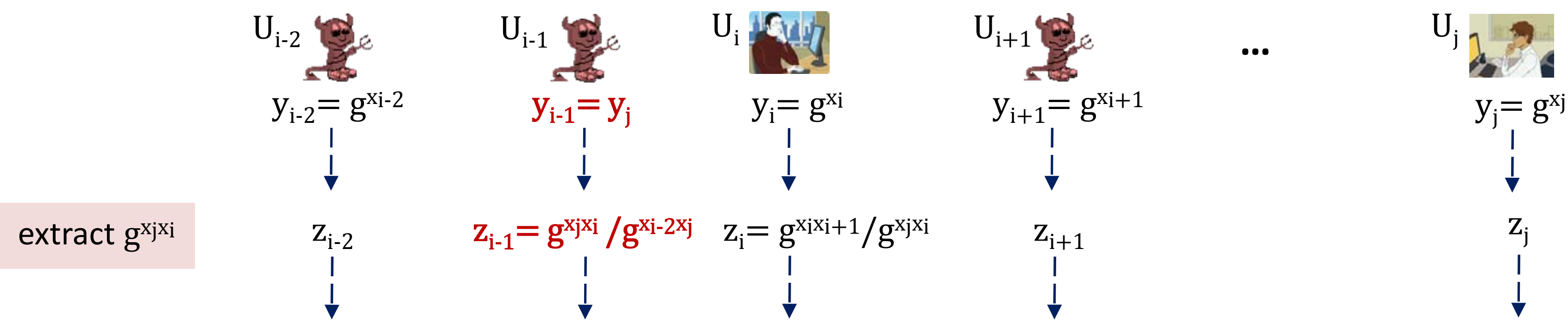
Each U_i broadcasts $z_i = (y_{i+1}/y_{i-1})^{x_i} = g^{x_ix_{i+1} - x_{i-1}x_i}$.

Each U_{i-1} can compute $k'_{i,i+1} = g^{x_ix_{i+1}}$ and each U_{i+1} can compute $k'_{i-1,i} = g^{x_{i-1}x_i}$.

Collusion Attacks

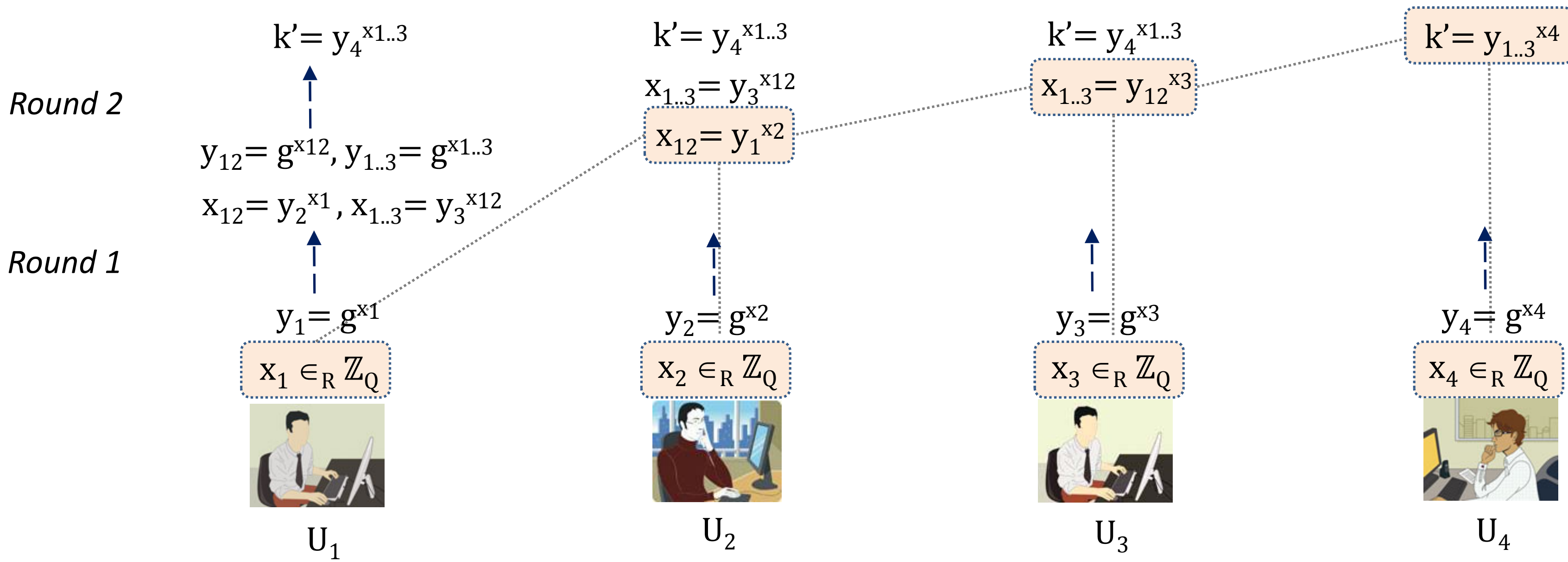
Any $k'_{i,i+1} = g^{x_ix_{i+1}}$ can be recovered through a collusion of U_j, $j \neq i$, $j \neq i+1$ from k'.

Any $k'_{i,j} = g^{x_ix_j}$ can be computed as follows:



Kim-Perrig-Tsudik GroupDH Protocol

Cyclic group $\mathbb{G} = (g, P, Q)$ s.t. if $x \in \mathbb{Z}_Q$ then $g^x \in \mathbb{Z}_Q$ (there is a bijection from $\mathbb{G}$ to $\mathbb{Z}_Q$).
$U_1, \dots, U_N$ are arranged as leaf nodes of a *full linear binary tree*.



Group DH element $k'_i = g^{x_N g^{x_{N-1} g^{\dots g^{x_3 g^{x_1 x_2}}}}}$

Analysis of PDHKE-KPT

Group key $k_i = H_g(g^{x_N g^{x_{N-1} g^{\cdots g^{x_3 g^{x_1 x_2}}}}}, (U_1, y_1), \ldots, (U_N, y_N))$

Pairwise key $k_{i,j} = H_p(g^{x_i x_j}, (U_i, y_i), (U_j, y_j))$

Is this secure?
Yes.

Observation

The only $k'_{i,j} = g^{x_i x_j}$ which appears in computations is $k'_{1,2} = g^{x_1 x_2}$.

But $k'_{1,2}$ is computed only by U_1 and U_2 **which is fine!**

Message $y_{1,2} = g^{k'_{1,2}}$ hides $k'_{1,2}$ in the exponent (hardness of DL).

Result

In ROM PDHKE-KPT is (passively) secure under the DDH and DL assumptions in $\mathbb{G}$.

<u>Intuition</u>

$y_{1,2} = g^{k'_{1,2}}$ is indistinguishable from $y^*_{1,2} \in_R \mathbb{G}$ under DDH assumption.

$k_{1,2} = H_p(g^{x_1 x_2}, (U_1, y_1), (U_2, y_2))$ is indistinguishable from $k^*_{1,2} \in_R \{0,1\}^\kappa$ unless $H_p(g^{x_1 x_2}, \ldots)$ is asked.

Authentication in GKE+P Protocols

Authentication Compiler for GKE Protocols (Katz-Yung'03)

uses EUF-CMA secure digital signature scheme $\Sigma = (\mathrm{KGen}(1^\kappa), \mathrm{Sig}(sk, m), \mathrm{Ver}(pk, m, \sigma))$

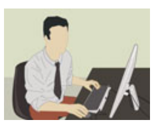
Katz-Yung'03: passive adversary = eavesdropper

Bresson-Manulis-Schwenk'07: passive adversary must be in the sense of Canetti-Krawczyk'01; otherwise insecure protocols exist

is also sufficient for authentication of passively secure GKE+P protocols

U_i

(sk_i, pk_i)



$r_i \in_R \{0,1\}^\kappa$ — U_i, r_i →

← $\{U_j, r_j\}_{j \neq i}$

$s_i = U_1|r_1|\ldots|U_N|r_N$

(passively) secure GKE+P protocol Π

m_{out} → $\sigma_{out} = \mathrm{Sig}(sk_i, (m_{out}, s_i))$ — $m_{out}|\sigma_{out}$ →

← m_{in} — if $\mathrm{Ver}(pk_j, (m_{in}, s_i), \sigma_{in})$ ← $m_{in}|\sigma_{in}$

k_i ↓ $k_{i,j}$ ↓

<u>Intuition</u>

Digital signatures on unique session ids prevent impersonation of messages exchanged between uncorrupted U_i and U_j (even if other parties are corrupted).

Conclusion

GKE+P protocol ⇒ 1 group key + up to N pairwise keys (on-demand w/o interaction)

New security challenges

independence between k and $k_{i,j}$

independence between $k_{i,j}$ and $k_{i,t}$ (also in the presence of collusions/insider adversaries)

Constructions

PDHKE with hash-based key derivation as a building block

exponent re-use technique in BD-PDHKE shown insecure, in KPT-PDHKE shown secure

authenticated GKE+P protocols can be obtained via Katz-Yung'03 authentication compiler for GKE

Not in the talk

Security model for GKE+P protocols (extension of Katz-Yung'03 model) and proofs

generic compiler from GroupDH to GKE+P based on PDHKE (can be extended for any GKE)

Open Question: What about Derivation of Subgroup Keys?

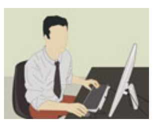
Generic Compilation of GKE+P Protocols

(passively) secure GroupDH protocol Π → compiler C → (passively) secure GKE+P protocol $\Pi' = C(\Pi)$

Compiler for GKE+P Protocols

Cyclic group $\mathbb{G} = (g, P, Q)$. Hash functions $H_g, H_p : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$.

U_i



$x_i \in_R \mathbb{Z}_Q$; $y_i = g^{x_i}$ — y_i →

← $\{y_j\}_{j \neq i}$

Π

k'_i ↓

$$k_i = H_g(k'_i, (U_1, y_1), \ldots, (U_N, y_N))$$

$$k_{i,j} = H_p(y_j^{x_i}, (U_i, y_i), \ldots, (U_j, y_j))$$

Remarks

Compiler is the combination of PDHKE and Π.

Exponents x_i used to compute $k'_{i,j}$ remain independent from x_i^* used in Π to compute k'_i.

If in Π each U_i broadcasts $y_i^* = g^{x_i^*}$ then y_i can be appended to y_i^* saving the preliminary round.

Independence of P2P Keys in PDHKE

yet we were considering indistinguishability of $k'_{i,j}$ from $k^* \in_R \mathbb{G}$

standard definitions require indistingushability from $k^* \in_R \{0,1\}^\kappa$

Key derivation and randomness extraction

<u>Hash Function</u>

$H : \{0,1\}^* \longrightarrow \{0,1\}^\kappa$. Good extractor in ROM (Bellare-Rogaway'93).

<u>Left-over-Hash-Lemma</u> (Håstad-Impagliazzo-Levin-Luby'99)

Based on universal hash functions, requires external perfect randomness.

<u>Truncation</u> (Chevalier-Fouque-Pointcheval-Zimmer'09)

Extract κ least significant bits. Good for DHKE-based protocols.

In PDHKE would additionally require PRF to admit further inputs.