Group Key Exchange Enabling On-Demand Derivation of P2P Keys

Mark Manulis
Cryptographic Protocols Group
TU Darmstadt & CASED
Users in $\mathcal{U} = \{U_1, \ldots, U_N\}$ run a **Group Key Exchange (GKE)** Protocol and compute a session group key $k$ indistinguishable from $k^\ast \in_R \{0,1\}^\kappa$.

A nice building block for group applications.

secure (private and authenticated) group channel for $U_1, \ldots, U_N$
Main Goal: Extending GKE with P2P Keys

One protocol $\Rightarrow$ **1 group key** + up to N peer-2-peer keys.
All keys must be independent (across different sessions).
Denote such protocols **GKE+P**.

**Naive solutions**

1. Execute GKE within $U$ 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 $U$ 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 extensions of 2-party DHKE (Diffie-Hellman‘76) to a group setting

**GroupDH**
is a GKE protocol amongst the users in $U = \{U_1, ..., U_N\}$ in which each $U_i$ chooses
own exponent $x_i \in_R \mathbb{Z}_Q$ and computes $k'_i = f(g, x_1, ..., x_N)$ 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 $Q$

$U_1$

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

$y_1 = g^{x_1}$

accept $k' = y_2^{x_1}$

$D_1$

$y_1$

$y_2 = g^{x_2}$

accept $k' = y_1^{x_2}$

$k' = g^{x_1x_2}$

secure against eavesdropping attacks under the DDH assumption

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

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

---

Group Key Exchange Enabling On-Demand Derivation of P2P Keys
ACNS 2009, Paris-Rocquencourt 02.06.2009 | Mark Manulis | www.manulis.eu
Parallel Diffie-Hellman Key Exchange

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

**PDHKE**

Let $U_1, U_i, U_N$ be a set of users (their unique identities).

$y_1 = g^{x_1}$

$y_1 \leftarrow$ accept $\{k'_{1,j} = y_1^{x_1}\}_j$

$y_i = g^{x_i}$

$y_i \leftarrow$ accept $\{k'_{i,j} = y_j^{x_i}\}_j$

$y_N = g^{x_N}$

$y_N \leftarrow$ accept $\{k'_{N,j} = y_j^{x_N}\}_j$

$U_i$ 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
but 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 $k^* \in_R G$
$U_i$ and $U_j$ are uncorrupted upon computation of $k'_{i,j}$ but any other $U$ can be corrupted

**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 $\{k'_{i,1} = g^{x_{ij}x_2}\}_i = \{k'_{i,2} = g^{x_{ij}x_2}\}_i$

$\downarrow$

A can distinguish any $k'_{i,2} = g^{x_{ij}x_2}$ from $k^*$ by revealing $k'_{i,1}$ from $U_i$
P2P Key Derivation in PDHKE

\[ U = \{U_1, \ldots, U_N\}. \] Hash function \( H_p : \{0,1\}^* \rightarrow \{0,1\}^\kappa \). 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**

- For each pair \((U_i, U_j)\): 
  \[ k_{i,j} = H_p(k'_{i,j}, (U_i, y_i), (U_j, y_j)) \]

- 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 $U = \{U_1, ..., U_N\}$ 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)
Merge GroupDH with PDHKE

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

Suitable key derivation
Hash functions \( H_g, H_p : \{0,1\}^* \rightarrow \{0,1\}^\kappa \). 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 $G = (g, P, Q)$. $U_1, ..., U_N$ are arranged into a cycle s.t. $U_0 = U_N$, $U_{N+1} = U_1$.

\[
\begin{align*}
U_{i-1} & \quad x_{i-1} \in \mathbb{Z}_Q \\
y_{i-1} &= g^{x_{i-1}} \\
z_{i-1} &= (y_i/y_{i-2})^{x_{i-1}} \\
U_i & \quad x_i \in \mathbb{Z}_Q \\
y_i &= g^{x_i} \\
z_i &= (y_{i+1}/y_i)^{x_i} \\
U_{i+1} & \quad x_{i+1} \in \mathbb{Z}_Q \\
y_{i+1} &= g^{x_{i+1}} \\
z_{i+1} &= (y_{i+2}/y_i)^{x_{i+1}}
\end{align*}
\]

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

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

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 \[ k_{ij} = H_p(g^{xi}, (U_i, y_i), (U_j, y_j)) \]

**Individual Attacks**

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

Each \( U_{i-1} \) can compute \( k'_{i,i+1} = g^{xi}x_{i+1} \) and each \( U_{i+1} \) can compute \( k'_{i-1,i} = g^{xi}x_{i-1} \).

**Collusion Attacks**

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

Any \( k'_{ij} = g^{xi}x_j \) can be computed as follows:

\[ y_i = g^x \quad y_{i-2} = g^{x_{i-2}} \quad z_{i-2} = g^{x_{i-2}}/g^{x_{i-2}} \quad z_1 = g^{x_{i+1}}/g^{x_{i-1}} \quad z_{i+1} = g^{x_{i+1}}/g^{x_{i-1}} \]

**Is this secure? No.**
Kim-Perrig-Tsudik GroupDH Protocol

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

Round 1

$y_1 = g^{x_1}$

$y_2 = g^{x_2}$

$y_3 = g^{x_3}$

$y_4 = g^{x_4}$

$U_1, U_2, U_3, U_4$

Round 2

$k' = y_4^{x_1 \cdot 3}$

$k' = y_4^{x_1 \cdot 3}$

$k' = y_4^{x_1 \cdot 3}$

$k' = y_4^{x_1 \cdot 4}$

$y_{12} = g^{x_{12}}, y_{13} = g^{x_{13}}$

$x_{12} = y_1^{x_1}, x_{13} = y_2^{x_1}$

$x_{12} = y_3^{x_1}, x_{13} = y_3^{x_1}$

$x_{12} = y_4^{x_1}, x_{13} = y_4^{x_1}$

$x_{12} \in \mathbb{R} \mathbb{Z}_Q, x_{13} \in \mathbb{R} \mathbb{Z}_Q, x_N \in \mathbb{R} \mathbb{Z}_Q$

$x_{N-1} g x_N g x_{N-2} g ...

x_2 g x_1 g x_2$

Group DH element $k'_1 = g^{x_N} g^{x_{N-1}} g ...

x_3 g x_1 g x_2$
Analysis of PDHKE-KPT

Group key  \[ k_i = H_g(g^{x_i}g^{x_{N-1}}g^{x_3}g^{x_1x_2}, (U_1, y_1), ..., (U_N, y_N)) \]

Pairwise key  \[ k_{ij} = H_p(g^{x_ix_j}, (U_i, y_i), (U_j, y_j)) \]

Observation

The only \( k'_{i,j} = g^{x_ix_j} \) which appears in computations is \( k'_{1,2} = g^{x_1x_2} \).

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 \( G \).

Intuition

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

\( k_{1,2} = H_p(g^{x_1x_2}, (U_1, y_1), (U_2, y_2)) \) is indistinguishable from \( k^*_{1,2} \in_R \{0,1\}^\kappa \) unless \( H_p(g^{x_1x_2}, ...) \) is asked.

Is this secure?
Yes.
Authentication in GKE+P Protocols

Authentication Compiler for GKE Protocols (Katz-Yung’03)
uses EUF-CMA secure digital signature scheme \( \Sigma = (\text{KGen}(1^\kappa), \text{Sig}(sk, m), \text{Ver}(pk, m, \sigma)) \)
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

\( (sk_i, pk_i) \)  
\( r_i \in_R \{0,1\}^\kappa \)  
\( s_i = U_1 | r_1 | ... | U_N | r_N \)  
\( (\text{passively}) \) secure GKE+P protocol \( \Pi \)  
\( m_{\text{out}} \)  
\( \sigma_{\text{out}} = \text{Sig}(sk_{\mu}, (m_{\text{out}}, s_{\mu})) \)  
\( m_{\text{in}} \)  
\( m_{\text{in}}, \sigma_{\text{in}} \)  
\( \text{Intuition} \)  
Digital signatures on unique session ids prevent impersonation of messages exchanged between uncorrupted \( U_i \) and \( U_j \) (even if other parties are corrupted).

\( k_i \)  
\( k_{ij} \)
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

Compiler for GKE+P Protocols
Cyclic group $\mathbb{G} = (g, P, Q)$. Hash functions $H_g, H_p : \{0,1\}^* \rightarrow \{0,1\}^k$.

\( U_i \) 
\[ x_i \in \mathbb{Z}_Q : y_i = g^{x_i} \]
\[ y_i \]
\[ \{y_j\}_{j \neq i} \]
\[ \Pi \]
\[ k'_i \downarrow \]
\[ k_i = H_g(k'_i, (U_1, y_1), ..., (U_N, y_N)) \]
\[ k_{ij} = H_p(y_i^{x_i}, (U_p, y_i), ..., (U_p, y_j)) \]

Remarks
Compiler is the combination of PDHKE and $\Pi$.

Exponents $x_i$ used to compute $k'_{ij}$ remain independent from $x_i^*$ used in $\Pi$ to compute $k'_i$.

If in $\Pi$ 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 $k^* \in_R G$

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

Key derivation and randomness extraction

Hash Function
$H : \{0,1\}^* \rightarrow \{0,1\}^\kappa$. Good extractor in ROM (Bellare-Rogaway’93).

Left-over-Hash-Lemma (Håstad-Impagliazzo-Levin-Luby’99)
Based on universal hash functions, requires external perfect randomness.

Truncation (Chevalier-Fouque-Pointcheval-Zimmer’09)
Extract $\kappa$ least significant bits. Good for DHKE-based protocols.
In PDHKE would additionally require PRF to admit further inputs.