Multi-Designated Receiver
Signed Public Key Encryption

Ueli Maurer, Christopher Portmann, Guilherme Rito
Eurocrypt 2022
Outline

1. Contributions

2. Public Key Encryption for Broadcast (PKEBC)

3. PKEBC Scheme Construction

4. Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE)

5. MDRS-PKE Scheme Construction
1. Contributions

- New types of Public Key Encryption (PKE) schemes:
  - Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE);
  - Public Key Encryption for Broadcast (PKEBC);
- Constructions of both schemes from standard assumptions;
- MDRS-PKE scheme yields Multi-Designated Verifier Signature scheme (MDVS) with Privacy of Identities;
- The only prior construction is based on Verifiable Functional Encryption [1].
1. Contributions

New types of Public Key Encryption (PKE) schemes:

- Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE);
- Public Key Encryption for Broadcast (PKEBC);
1. Contributions

New types of Public Key Encryption (PKE) schemes:

• Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE);
• Public Key Encryption for Broadcast (PKEBC);

Constructions of both schemes from standard assumptions;
1. Contributions

New types of Public Key Encryption (PKE) schemes:

- Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE);
- Public Key Encryption for Broadcast (PKEBC);

Constructions of both schemes from standard assumptions;

MDRS-PKE scheme yields Multi-Designated Verifier Signature scheme (MDVS) with Privacy of Identities;

The only prior construction is based on Verifiable Functional Encryption [1].

[1] Damgård et al. (TCC ’20).
Outline

1. Contributions

2. Public Key Encryption for Broadcast (PKEBC)

3. PKEBC Scheme Construction

4. Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE)

5. MDRS-PKE Scheme Construction
2. PKEBC — Syntax

\[
\text{Syntax of decryption:}
\begin{itemize}
\item Only \(\text{rsk}_{B1}\) is needed;
\item Outputs vector of receivers' \(\text{rpk}\).
\end{itemize}
\]
2. PKEBC — Syntax

\[ c = E((r_{pk}B1, r_{pk}B4, r_{pk}B3), m) \]
2. PKEBC — Syntax

Syntax of decryption:

• Only $rsk_{B1}$ is needed;
• Outputs vector of receivers' $rpk$.
2. PKEBC — Syntax

\[ c = E((rpk_{B1}, rpk_{B4}, rpk_{B3}), m) \]

Syntax of decryption:

• Only \( rsk_{B1} \) is needed;
• Outputs vector of receivers’ \( rpk \).
2. PKEBC — Syntax

Mathematical expression:
\[ c = E((rpk_{B1}, rpk_{B4}, rpk_{B3}), m) \]

Syntax of decryption:
- Only \( rsk_{B1} \) is needed;
- Outputs vector of receivers' \( rpk \).
2. PKEBC — Syntax

\[ c = E((rpk_{B1}, rpk_{B4}, rpk_{B3}), m) \]

Syntex of decryption:
• Only rsk_{B1} is needed;
• Outputs vector of receivers’ rpk.

\[ D(rsk_{B1}, c) = \]
2. PKEBC — Syntax

\[ c = E((rpk_{B1}, rpk_{B4}, rpk_{B3}), m) \]

\[ D(rsk_{B1}, c) = ((rpk_{B1}, rpk_{B4}, rpk_{B3}), m) \]
2. PKEBC — Syntax

Syntax of decryption:

- Only \( rsk_{B1} \) is needed;
- Outputs vector of receivers’ \( rpk \).
2. PKEBC — Security Notions

Consistency

Robustness

Confidentiality (IND-CCA-2 security)
2. PKEBC — Security Notions

Consistency

Robustness

Confidentiality (IND-CCA-2 security)

+ Anonymity (IK-CCA-2 security)
2. PKEBC — Consistency

\[
D (rsk_{B1}, c) = (rpk_{B1}, rpk_{B3}, rpk_{B5}), m
\]

\[
D (rsk_{B3}, c) = (rpk_{B1}, rpk_{B3}, rpk_{B5}), m
\]
2. PKEBC — Consistency

\[ D_{sk_{B1}, c}(rpk_{B1}, rpk_{B3}, rpk_{B5}) = (m, m) \]

\[ D_{sk_{B3}, c}(rpk_{B1}, rpk_{B3}, rpk_{B5}) = (m, m) \]
2. PKEBC — Consistency

\[ D(c_{B1}, c) = ((r_{pk}^{B1}, r_{pk}^{B3}, r_{pk}^{B5}), m) \]

\[ D(c_{B3}, c) = ((r_{pk}^{B1}, r_{pk}^{B3}, r_{pk}^{B5}), m) \]
2. PKEBC — Consistency

\[
D((rsk_{B1}, c)) = ((rpk_{B1}, rpk_{B3}, rpk_{B5}), m)
\]

\[
D((rsk_{B3}, c)) = ((rpk_{B1}, rpk_{B3}, rpk_{B5}), m)
\]
2. PKEBC — Consistency

\[ D(\text{rsk}_{B1}, c) = ((\text{rpk}_{B1}, \text{rpk}_{B3}, \text{rpk}_{B5}), m) \]
2. PKEBC — Consistency

\[ D(rsk_{B1}, c) = ((rpk_{B1}, rpk_{B3}, rpk_{B5}), m) \]

\[ D(rsk_{B3}, c) = ((rpk_{B1}, rpk_{B3}, rpk_{B5}), m) \]
2. PKEBC — Robustness
2. PKEBC — Robustness

\[ c = E((r_{pk}^{B1}, r_{pk}^{B4}), m) \]

\[ D(r_{sk}^{B3}, c) = (r_{pk}^{B1}, r_{pk}^{B4}) \]

\[ D(r_{sk}^{B3}, c) = \bot \]
2. PKEBC — Robustness

\[ c = E((rpk_{B1}, rpk_{B4}), m) \]
2. PKEBC — Robustness

\[ c = E((\text{rpk}_{B1}, \text{rpk}_{B4}), m) \]

\[ D(\text{rsk}_{B1}, c) = ((\text{rpk}_{B1}, \text{rpk}_{B4}), m) \]
2. PKEBC — Robustness

\[ c = E((rpk_{B1}, rpk_{B4}), m) \]

\[ D(rsk_{B1}, c) = ((rpk_{B1}, rpk_{B4}), m) \]
2. PKEBC — Robustness

\[ c = E((rpk_{B1}, rpk_{B4}), m) \]

\[ D(rsk_{B1}, c) = ((rpk_{B1}, rpk_{B4}), m) \]

\[ D(rsk_{B3}, c) = \bot \]
2. PKEBC — Confidentiality (IND-CCA-2 Security)

\[ c = E((rpk_{B_1}, rpk_{B_4}), m) \]
2. PKEBC — Confidentiality (IND-CCA-2 Security)

\[ c = E((rpk_{B1}, rpk_{B4}), m) \]
2. PKEBC — Confidentiality (IND-CCA-2 Security)

\[ c = E((rpk_{B1}, rpk_{B4}), m) \]

\[ m = ? \]
2. PKEBC — Anonymity (IK-CCA-2 Security)

\[ c = E((rpk_{B1}, rpk_{B4}), m) \]
2. PKEBC — Anonymity (IK-CCA-2 Security)

\[ c = E((\text{rpkB}_1, \text{rpkB}_4), m) \]

Are the (two) receivers of \( c \) \((B1, B4)\)?
2. PKEBC — Anonymity (IK-CCA-2 Security)

c = E((rpk_{B2}, rpk_{B4}), m)

Are the (two) receivers of \( c \) \( (B2, B4) \)?
2. PKEBC — Anonymity (IK-CCA-2 Security)

A

\[ c = E((\text{rpk}_{B2}, \text{rpk}_{B3}), m) \]

Are the (two) receivers of \(c\) \((B2, B3)\)?

B1

B2

B3

B4

B5
2. PKEBC — Anonymity (IK-CCA-2 Security)

\[ c = E((rpk_{B3}, rpk_{B4}), m) \]

Are the (two) receivers of \( c (B3, B4) \)?
Outline

1. Contributions

2. Public Key Encryption for Broadcast (PKEBC)

3. PKEBC Scheme Construction

4. Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE)

5. MDRS-PKE Scheme Construction
3. PKEBC Scheme Construction Roadmap

Recall Naor-Yung’s IND-CCA-1 PKE scheme construction [2];

Generalization to (non IK-CCA-2 secure) PKEBC scheme;

Making the PKEBC scheme IK-CCA-2 secure.

3. Naor-Yung’s PKE Scheme

Building blocks:

- (IND-CPA secure) PKE scheme CPA = (Gen, Enc, Dec);
- NIZK = (Gen_{CRS}, Prv, Vfy);
3. Naor-Yung’s PKE Scheme

Building blocks:
- (IND-CPA secure) PKE scheme $\text{CPA} = (\text{Gen}, \text{Enc}, \text{Dec})$;
- $\text{NIZK} = (\text{Gen}_{\text{CRS}}, \text{Prv}, \text{Vfy})$;

Construction of IND-CCA-1 secure PKE scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$:

$\Pi.\text{Gen}$:

\[
\left((pk_0, sk_0), (pk_1, sk_1)\right) \leftarrow (\text{CPA.Gen, CPA.Gen});
\]

\[
crs \leftarrow \text{NIZK.Gen}_{\text{CRS}};
\]

Output $\left(pk := (crs, pk_0, pk_1), sk := (sk_0, pk)\right)$. 
3. Naor-Yung’s PKE Scheme

\[ \Pi.\text{Enc}(pk := (\text{crs}, pk_0, pk_1), m): \]
\[ (c_0, c_1) \leftarrow (\text{CPA.Enc}_{pk_0}(m), \text{CPA.Enc}_{pk_1}(m)); \]
\[ p \leftarrow \text{NIZK.Prv}(\text{crs}, \text{stmt} := \text{“There is a message } m \text{ such that } c_0 \text{ and } c_1 \text{ are encryptions of } m \text{ under } pk_0 \]
\[ \text{and } pk_1, \text{ resp.”}, \]
\[ w := (m, \text{Encryption Randomness}); \]
\[ \text{Output } (p, c_0, c_1). \]

Simulation Sound NIZK $\Rightarrow$ PKE scheme is IND-CCA-2 secure [3].
3. Naor-Yung’s PKE Scheme

Π.Enc(pk := (crs, pk₀, pk₁), m):
   (c₀, c₁) ← (CPA.Enc(pk₀)(m), CPA.Enc(pk₁)(m));
   p ← NIZK.Prv(crs,
   stmt := “There is a message m such that c₀ and c₁ are encryptions of m under pk₀
   and pk₁, resp.”,
   w := (m, Encryption Randomness));
   Output (p, c₀, c₁).

Π.Dec(sk := (sk₀, pk), c := (p, c₀, c₁)):
   Output ⊥ if NIZK.Vfy(crs, stmt, p) = invalid;
   m ← CPA.Dec(sk₀)(c₀);
   Output m otherwise.
3. Naor-Yung’s PKE Scheme

\(\Pi.\text{Enc}(pk := (\text{crs}, pk_0, pk_1), m):\)

\[(c_0, c_1) \leftarrow (\text{CPA.Enc}_{pk_0}(m), \text{CPA.Enc}_{pk_1}(m));\]

\[p \leftarrow \text{NIZK.Prv}(\text{crs},\]

\[\text{stmt} := \text{“There is a message } m \text{ such that } c_0 \text{ and } c_1 \text{ are encryptions of } m \text{ under } pk_0 \]

\[\text{and } pk_1, \text{ resp.”,}\]

\[w := (m, \text{Encryption Randomness});\]

\[\text{Output } (p, c_0, c_1).\]

\(\Pi.\text{Dec}(sk := (sk_0, pk), c := (p, c_0, c_1)):\)

\[\text{Output } \bot \text{ if } \text{NIZK.Vfy}(\text{crs, stmt, } p) = \text{invalid;}\]

\[m \leftarrow \text{CPA.Dec}_{sk_0}(c_0);\]

\[\text{Output } m \text{ otherwise.}\]

Simulation Sound NIZK \(\Rightarrow\) PKE scheme is IND-CCA-2 secure [3].

[3]: Sahai, FOCS ’99
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)

\[ \Pi. \text{Setup}: \]

\[ \text{crs} \leftarrow \text{NIZK.Gen}_{\text{CRS}}; \]
\[ \text{Output crs}; \]

\[ \Pi. \text{Gen}: \]

\[ \left( (pk_0, sk_0), (pk_1, sk_1) \right) \leftarrow (\text{CPA.Gen}, \text{CPA.Gen}); \]
\[ \text{crs} \leftarrow \text{NIZK.Gen}_{\text{CRS}}; \]
\[ \text{Output} \left( rpk := (pk_0, pk_1), rsk := (sk_0, rpk) \right). \]
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)

Π.Enc(pp := crs, \vec{v} := (rpk_1 := (pk_{1,0}, pk_{1,1}), \ldots, rpk_{|\vec{v}|} := (pk_{|\vec{v}|,0}, pk_{|\vec{v}|,1})), m):
(c_j,0, c_j,1) \leftarrow (\text{CPA.Enc}_{pk_j,0}(m), \text{CPA.Enc}_{pk_j,1}(m)), \text{ for each } j \in \{1, \ldots, |\vec{v}|\}

p \leftarrow \text{NIZK.Prv}(crs,
\text{stmt} := \text{"There is a message } m \text{ such that for all } j \in \{1, \ldots, |\vec{v}|\}, \text{ and all } b \in \{0, 1\}, c_{j,b} \text{ is an encryption of } m \text{ under } v_{j,b}.",
\text{w} := (m, \text{Encryption Randomness})
)
Output (p, \vec{c} := ((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{v}|,0}, c_{|\vec{v}|,1})), \vec{v}).
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)

\[ \Pi.\text{Enc}(pp := \text{crs}, \vec{v} := (rpk_1 := (pk_{1,0}, pk_{1,1}), \ldots, rpk_{|\vec{v}|} := (pk_{|\vec{v}|,0}, pk_{|\vec{v}|,1})), m) :=
\]
\[ (c_j,0, c_j,1) \leftarrow (\text{CPA.Enc}_{pk_j,0}(m), \text{CPA.Enc}_{pk_j,1}(m)), \text{for each } j \in \{1, \ldots, |\vec{v}|\}
\]
\[ p \leftarrow \text{NIZK.Prv}(\text{crs},
\]
\[ \text{stmt} := “\text{There is a message } m \text{ such that for all } j \in \{1, \ldots, |\vec{v}|\}, \text{and all } b \in \{0, 1\},
\]
\[ c_{j,b} \text{ is an encryption of } m \text{ under } v_{j,b}.”,
\]
\[ w := (m, \text{Encryption Randomness})
\]
\[ \text{Output } (p, \vec{c} := ((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{v}|,0}, c_{|\vec{v}|,1})), \vec{v}).
\]

\[ \Pi.\text{Dec}(pp := \text{crs, rsk := (sk_0, rpk)}, c := (p, \vec{c} := ((c_{1,0}, c_{1,1}), \ldots, (c_{|\vec{v}|,0}, c_{|\vec{v}|,1})), \vec{v})) :=
\]
\[ \text{Output } \bot \text{ if } \text{NIZK.Vfy}(\text{crs, stmt, p}) = \text{invalid};
\]
\[ \text{Let } i \in \{1, \ldots, |\vec{v}|\} \text{ be (the least number) such that } v_{i} = rpk;
\]
\[ \text{Output } \bot \text{ if there is no such } i;
\]
\[ m \leftarrow \text{CPA.Dec}_{sk_0}(c_i,0);
\]
\[ \text{Output } (\vec{v}, m) \text{ otherwise.}
\]
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)

- PKE (IND-CPA)
- NIZK
- Simulation Sound

- Exists from Standard Assumptions
- Does not exist from Standard Assumptions

Eurocrypt' 22 16/35
3. Naor-Yung Scheme’s Trivial Generalization (Non IK-CCA-2 Secure)
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

Main idea:
Add a (Binding) Commitment to $\vec{v}$ (vector of receivers' public keys) and $m$; Encrypt $\vec{v}$, $m$ and commitment's randomness to each receiver;
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

Main idea:

Add a (Binding) Commitment to $\vec{v}$ (vector of receivers’ public keys) and $m$;
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

Main idea:

Add a (Binding) Commitment to $\vec{v}$ (vector of receivers’ public keys) and $m$;

Encrypt $\vec{v}$, $m$ and commitment’s randomness to each receiver;
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

Building Blocks:

- (Statistically Binding) Commitment scheme $CS = (Gen_{CRS}, \text{Commit}, \text{Verify})$;
- (IND-CPA and IK-CPA secure) PKE scheme $CPA = (Gen, \text{Enc}, \text{Dec})$;
- (Simulation Sound) NIZK $= (Gen_{CRS}, \text{Prv}, \text{Vfy})$. 
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

\[\Pi\).

\textbf{Setup:}
\[
\text{crs}_{\text{NIZK}} \leftarrow \text{NIZK.Gen}_{\text{CRS}};
\]
\[
\text{crs}_{\text{CS}} \leftarrow \text{CS.Gen}_{\text{CRS}};
\]
\text{Output } (\text{crs}_{\text{NIZK}}, \text{crs}_{\text{CS}});
\]

\[\Pi\).

\textbf{Gen:}
\[
\left((\text{pk}_0, \text{sk}_0), (\text{pk}_1, \text{sk}_1)\right) \leftarrow (\text{CPA.Gen, CPA.Gen});
\]
\text{Output } \left(\text{rpk} := (\text{pk}_0, \text{pk}_1), \text{rsk} := (\text{sk}_0, \text{rpk})\right).
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

\[ \Pi.\text{Enc}(pp := (\text{crs}_\text{NIZK}, \text{crs}_\text{CS}), \vec{v} := (\text{rpk}_0 := (\text{pk}_{1,0}, \text{pk}_{1,1}), \ldots, \text{rpk}_{|\vec{v}|} := (\text{pk}_{|\vec{v}|,0}, \text{pk}_{|\vec{v}|,1}), m) : \\
\text{comm} \leftarrow \text{CS.Commit}(\text{crs}_\text{CS}, (\vec{v}, m); \rho); \\
(c_{j,0}, c_{j,1}) \leftarrow (\text{CPA.Enc}_{\text{pk},0}(\rho, \vec{v}, m), \text{CPA.Enc}_{\text{pk},1}(\rho, \vec{v}, m)), \text{for each } i \in \{1, \ldots, |\vec{v}|\}; \\
p \leftarrow \text{NIZK.Prv}(\text{crs}_\text{NIZK}, \\
\text{stmt} := \"There is a message } m, \text{ a vector } \vec{v} \text{ and a sequence } \rho \text{ such that:} \\
\text{for all } i \in \{1, \ldots, |\vec{v}|\}, b \in \{0, 1\}, c_{i,b} \text{ is an encryption of } (\rho, \vec{v}, m) \text{ under } v_{i,b}, \\
\text{and } \text{comm} = \text{CS.Commit}(\text{crs}_\text{CS}, (\vec{v}, m); \rho).\"; \\
w := (m, \vec{v}, \rho, \text{Encryption Randomness}); \\
\text{Output } (\text{comm}, p, \vec{c}). \]
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

\[ \Pi.\text{Dec}(pp := (\text{crs}_{\text{NIZK}}, \text{crs}_{\text{CS}}),\text{rsk} := (sk_0, rpk), c := (\text{comm}, p, \vec{c})):} \]

Output \( \bot \) if NIZK.Vfy(\text{crs}, \text{stmt}, p) = \text{invalid};

Find the least \( i \in \{1, \ldots, |\vec{v}|\} \) with

\[(\rho, \vec{v}, m) \leftarrow \text{CPA.}\text{Dec}_{sk_0}(c_i, 0) \text{ satisfying:}
\]

\[(\rho, \vec{v}, m) \neq \bot; \]

\[v_i = rpk;\]

\[c.\text{comm} = \text{CS.}\text{Commit}(\text{crs}_{\text{CS}}, (\vec{v}, m); \rho);\]

Output \( \bot \) if there is no such \( i \);

Output \((\vec{v}, m)\) otherwise.
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

- PKE (IND-CPA)
- Simulation Sound
- NIZK
- From [2]
- PKEBC

From [2]
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

PKE (IND-CPA) \rightarrow NIZK (Simulation Sound) \rightarrow Statistically Binding Commitment Scheme

From [2]
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

From [2]

- IK-CPA
- PKE (IND-CPA)
- Simulation Sound NIZK
- Statistically Binding Commitment Scheme

PKEBC
3. Our (IK-CCA-2 Secure) PKEBC Scheme Construction

- IK-CPA PKE (IND-CPA)
- Simulation Sound NIZK
- Statistically Binding Commitment Scheme
- From [2]
- Ours
- Anonymous PKEBC
Outline

1. Contributions

2. Public Key Encryption for Broadcast (PKEBC)

3. PKEBC Scheme Construction

4. Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE)

5. MDRS-PKE Scheme Construction
4. MDRS-PKE — Syntax

Syntax of decryption:
• Only \( rsk_{B1} \) is needed;
• Outputs sender's \( spk \) and vector of receivers' \( rpk \).

\[
c = E(spk_{A1}, (rpk_{B1}, rpk_{B2}), m) \\
D(rsk_{B1}, c) = (spk_{A1}, (rpk_{B1}, rpk_{B2}), m)
\]
4. MDRS-PKE — Syntax

Syntax of decryption:
• Only \(rsk_{B1}\) is needed;
• Outputs sender's \(spk\) and vector of receivers' \(rpk\).
4. MDRS-PKE — Syntax

\[ \begin{align*}
(r_{pk}B_1, r_{sk}B_1) \\
(r_{pk}B_2, r_{sk}B_2) \\
(r_{pk}B_3, r_{sk}B_3) \\
(r_{pk}B_4, r_{sk}B_4) \\
(r_{pk}B_5, r_{sk}B_5)
\end{align*} \]

\[ \begin{align*}
(s_{pk}A_1, s_{sk}A_1) \\
(s_{pk}A_2, s_{sk}A_2) \\
(s_{pk}A_3, s_{sk}A_3)
\end{align*} \]

Syntax of decryption:
• Only \( r_{sk}B_1 \) is needed;
• Outputs sender’s \( s_{pk} \) and vector of receivers’ \( r_{pk} \).

\[ c = E(s_{sk}A_1, (r_{pk}B_1, r_{pk}B_2)), m \]
\[ D(r_{sk}B_1, c) = (s_{pk}A_1, (r_{pk}B_1, r_{pk}B_2)), m \]

Eurocrypt’ 22 24/35
4. MDRS-PKE — Syntax

\[ c = E(ssk_{A1}, (rpk_{B1}, rpk_{B2}), m) \]

\[ (rpk_{B1}, rsk_{B1}) \]

\[ (rpk_{B2}, rsk_{B2}) \]

\[ (rpk_{B3}, rsk_{B3}) \]

\[ (rpk_{B4}, rsk_{B4}) \]

\[ (rpk_{B5}, rsk_{B5}) \]

Syntax of decryption:

- Only \( rsk_{B1} \) is needed;
- Outputs sender's \( spk \) and vector of receivers' \( rpk \).
4. MDRS-PKE — Syntax

Syntax of decryption:

- Only $rsk_{B1}$ is needed;
- Outputs sender's $spk_{A1}$ and vector of receivers' $rpk_{B1, B2}$.

\[
D(rsk_{B1}, c) = \langle \text{spk}_{A1}, \langle rpk_{B1}, rpk_{B2} \rangle, m \rangle
\]
4. MDRS-PKE — Syntax

Syntax of decryption:

- Only $rsk_{B1}$ is needed;
- Outputs sender’s $spk$ and vector of receivers’ $rpk$.

$D(rsk_{B1}, c) = (spk_{A1}, (rpk_{B1}, rpk_{B2}), m) = (spk_{A2}, (rpk_{B1}, rpk_{B2}), m)$
4. MDRS-PKE — Security Notions

Off-The-Record

Unforgeability

Consistency

Confidentiality (IND-CCA-2 security)

+ Anonymity (IK-CCA-2 security)
4. MDRS-PKE — Off-The-Record

\[ c = \text{E}(\text{ssk}_{A2}, (\text{rpk}_{B4}, \text{rpk}_{B5}, \text{rpk}_{B2}), m) \]
4. MDRS-PKE — Off-The-Record

Is the sender $A_2$ and the two receivers ($B_1, B_4$)?

$$D(rsk_{B_5}, c) = (spk_{A_2}, (rpk_{B_4}, rpk_{B_5}, rpk_{B_2}), m)$$

$$c = E(ssk_{A_2}, (rpk_{B_4}, rpk_{B_5}, rpk_{B_2}), m)$$

Look, $A_2$ sent $m$ to ($B_4, B_5, B_3$)! Check it with my secret key: $rsk_{B_5}$.

ssk$_{A_2}$ not needed!

I'm not convinced!
4. MDRS-PKE — Off-The-Record

Is the sender A2 and the two receivers (B1, B4)?

\[ D(rsk_{B5}, c) = (spk_{A2}, (rpk_{B4}, rpk_{B5}, rpk_{B2}), m) \]

Look, A2 sent \( m \) to (B4, B5, B3)! Check it with my secret key: \( rsk_{B5} \).

\[ c = E(ssk_{A2}, (rpk_{B4}, rpk_{B5}, rpk_{B2}), m) \]
4. MDRS-PKE — Off-The-Record

Is the sender A2 and the two receivers (B1, B4)?

\[ D(rsk_{B3}, c) = \perp, \text{ or } D(rsk_{B3}, c) = (spk, \vec{v}, m) \]

and for every (honest) sender \( A_i \), either \( spk \neq spk_{A_i} \) or \( A_i \) sent \( m \) to \( \vec{v} \) before.

\[ D(rsk_{B5}, c) = (spk_{A2}, (rpk_{B4}, rpk_{B5}, rpk_{B2}), m) \]

Look, A2 sent \( m \) to (B4, B5, B3)! Check it with my secret key: \( rsk_{B5} \).

ssk_{A2} not needed!

I'm not convinced!
4. MDRS-PKE — Off-The-Record

Is the sender A2 and the two receivers (B1, B4)?

\[ D(rsk_{B3}, c) = \bot, \text{ or } D(rsk_{B3}, c) = (spk, \vec{v}, m) \]

and for every (honest) sender \( A_i \), either \( spk \neq spk_{A_i} \) or \( A_i \) sent \( m \) to \( \vec{v} \) before.

Look, A2 sent \( m \) to (B4, B5, B3)!
Check it with my secret key: \( rsk_{B5} \).

\[ c = \text{Forge}(rsk_{B5}, spk_{A2}, (rpk_{B4}, rpk_{B5}, rpk_{B2}), m) \]

\( ssk_{A2} \) not needed!

I'm not convinced!
4. MDRS-PKE — Off-The-Record

Is the sender A2 and the two receivers (B1, B4)?

Look, A2 sent m to (B4, B5, B3)! Check it with my secret key: rskB5.

sskA2 not needed!
4. MDRS-PKE — Off-The-Record

A2

A2 sent \( m \) to (B4, B5, B3)!
Check it with my secret key: \( rsk_{B5} \).

I'm not convinced!

c

ssk_{A2} not needed!
4. MDRS-PKE — Authenticity (Existential Unforgeability)

Is the sender A2 and the two receivers (B1, B4)?
4. MDRS-PKE — Authenticity (Existential Unforgeability)

Is the sender A2 and the two receivers (B1, B4)?

$D(rsk_{B3}, c) = \perp$, or $D(rsk_{B3}, c) = (spk, \vec{v}, m)$ and for every (honest) sender $A_i$, either $spk \neq spk_{A_i}$ or $A_i$ sent $m$ to $\vec{v}$ before.
Is the sender $A_2$ and the two receivers $(B_1, B_4)$?

$D(rsk_{B_3}, c) = \perp$, or $D(rsk_{B_3}, c) = (spk, \vec{v}, m)$ and for every (honest) sender $A_i$, either $spk \neq spk_{A_i}$ or $A_i$ sent $m$ to $\vec{v}$ before.
4. MDRS-PKE — Consistency

$D(rsk_{B3}, c) = \bot$, or $D(rsk_{B3}, c) = (spk, \vec{v}, m)$ and for every (honest) sender $A_i$, either $spk \neq spk_{A_i}$ or $A_i$ sent $m$ to $\vec{v}$ before.

Is the sender $A_2$ and the two receivers $(B_1, B_4)$?
4. MDRS-PKE — Consistency

Is the sender A2 and the two receivers (B1, B4)?

\[ D(\text{rsk}_{B1}, c) = (\text{spk}_X, (\text{rpk}_{B1}, \text{rpk}_{B4}, \text{rpk}_{B5}), m) \]

\[ D(\text{rsk}_{B4}, c) = (\text{spk}_X, (\text{rpk}_{B1}, \text{rpk}_{B4}, \text{rpk}_{B5}), m) \]
4. MDRS-PKE — Consistency

\[ D(\text{rsk}_{B_1}, c) = (\text{spk}_X, (\text{rpk}_{B_1}, \text{rpk}_{B_4}, \text{rpk}_{B_5}), m) \]
4. MDRS-PKE — Consistency

\[ D(rsk_{B1}, c) = (spk_X, (rpk_{B1}, rpk_{B4}, rpk_{B5}), m) \]

\[ D(rsk_{B4}, c) = (spk_X, (rpk_{B1}, rpk_{B4}, rpk_{B5}), m) \]
4. MDRS-PKE — Confidentiality (IND-CCA-2 Security)

Is the sender A2 and the two receivers (B1, B4)?
4. MDRS-PKE — Confidentiality (IND-CCA-2 Security)

Is the sender $A_2$ and the two receivers ($B_1$, $B_4$)?

$c \equiv E(\text{ssk}_{A_2}, (\text{rpk}_{B_1}, \text{rpk}_{B_4}), m)$
4. MDRS-PKE — Confidentiality (IND-CCA-2 Security)

Is the sender A2 and the two receivers (B1, B4)?

$D(rsk_{B3}, c) = \bot$, or $D(rsk_{B3}, c) = (spk, \vec{v}, m)$ and for every (honest) sender $A_i$, either $spk \neq spk_{A_i}$ or $A_i$ sent $m$ to $\vec{v}$ before.

$c = E(ssk_{A2}, (rpk_{B1}, rpk_{B4}), m)$
4. MDRS-PKE — Confidentiality (IND-CCA-2 Security)

Is the sender A2 and the two receivers (B1, B4)?

\[ c = E(ssk_{A2}, (rpk_{B1}, rpk_{B4}), m) \]

For every (honest) sender \( A_i \), either \( spk \neq spk_{A_i} \) or \( A_i \) sent \( m \) to \( \vec{v} \) before.

\[ m = ? \]
4. MDRS-PKE — Anonymity (IK-CCA-2 Security)

\[ c = E(\text{ssk}_{A2}, (\text{rpk}_{B1}, \text{rpk}_{B4}), m) \]
4. MDRS-PKE — Anonymity (IK-CCA-2 Security)

$c = E(\text{sk}_{A2}, (\text{rpk}_{B1}, \text{rpk}_{B4}), m)$

Is the sender A2 and the two receivers (B1, B4)?

Is the sender A2 and the two receivers (B1, B4)?

Is the sender A2 and the two receivers (B1, B4)?

Is the sender A2 and the two receivers (B1, B4)?

Is the sender A2 and the two receivers (B1, B4)?

Is the sender A2 and the two receivers (B1, B4)?

Is the sender A2 and the two receivers (B1, B4)?
4. MDRS-PKE — Anonymity (IK-CCA-2 Security)

\[ c = E(\text{ssk}_{A1}, (\text{rpk}_{B1}, \text{rpk}_{B4}), m) \]

Is the sender A1 and the two receivers (B1, B4)?

Is the sender A1 and the two receivers (B1, B3)?

Is the sender A1 and the two receivers (B1, B4)?

Is the sender A1 and the two receivers (B1, B4)?
4. MDRS-PKE — Anonymity (IK-CCA-2 Security)

\[ c = E(ssk_{A1}, (rpk_{B1}, rpk_{B3}), m) \]

Is the sender A1 and the two receivers (B1, B3)?

Is the sender A1 and the two receivers (B1, B3)?

Is the sender A2 and the two receivers (B2, B4)?

Is the sender A2 and the two receivers (B2, B4)?
4. MDRS-PKE — Anonymity (IK-CCA-2 Security)

\[ c = E(\text{ssk}_{A2}, (\text{rpk}_{B2}, \text{rpk}_{B4}), m) \]

Is the sender A2 and the two receivers (B2, B4)?

Is the sender A2 and the two receivers (B2, B4)?
Outline

1. Contributions

2. Public Key Encryption for Broadcast (PKEBC)

3. PKEBC Scheme Construction

4. Multi-Designated Receiver Signed Public Key Encryption (MDRS-PKE)

5. MDRS-PKE Scheme Construction
5. Our MDRS-PKE Scheme Construction

Building Blocks:

- \(\text{MDVS} = (\text{Setup}, \text{GenSig}, \text{GenVrf}, \text{Sign}, \text{Vfy})\)
- \((\text{IK-CCA-2 secure}) \text{PKEBC} = (\text{Setup}, \text{GenSnd}, \text{GenRcv}, \text{Enc}, \text{Dec})\)

Main idea: “Sign-then-Encrypt”

Use MDVS to sign vector of receivers and message; Then, use PKEBC to encrypt the sender’s and all receivers’ public keys, message and the signature.
5. Our MDRS-PKE Scheme Construction

Building Blocks:

- \( \text{MDVS} = (\text{Setup}, \text{Gen}_{\text{Sig}}, \text{Gen}_{\text{Vrf}}, \text{Sign}, \text{Vfy}) \);
- (IK-CCA-2 secure) \( \text{PKEBC} = (\text{Setup}, \text{Gen}_{\text{Snd}}, \text{Gen}_{\text{Rcv}}, \text{Enc}, \text{Dec}) \);

Main idea: "Sign-then-Encrypt"
Use MDVS to sign vector of receivers and message; Then, use PKEBC to encrypt the sender’s and all receivers’ public keys, message and the signature.
5. Our MDRS-PKE Scheme Construction

Building Blocks:

- MDVS = \((Setup, Gen_{Sig}, Gen_{Vrf}, Sign, Vfy)\);
- (IK-CCA-2 secure) PKEBC = \((Setup, Gen_{Snd}, Gen_{Rcv}, Enc, Dec)\);

Main idea: “Sign-then-Encrypt”
5. Our MDRS-PKE Scheme Construction

Building Blocks:

- MDVS = \( (\text{Setup}, \text{Gen}_{\text{Sig}}, \text{Gen}_{\text{Vrf}}, \text{Sign}, \text{Vfy}) \);
- (IK-CCA-2 secure) PKEBC = \( (\text{Setup}, \text{Gen}_{\text{Snd}}, \text{Gen}_{\text{Rcv}}, \text{Enc}, \text{Dec}) \);

Main idea: “Sign-then-Encrypt”

Use MDVS to sign vector of receivers and message;
Then, use PKEBC to encrypt the sender’s and all receivers’ public keys, message and the signature.
5. Our MDRS-PKE Scheme Construction

- NIZK Simulation Sound PKE (IND-CPA)
- Statistically Binding Commitment Scheme
- Anonymous Ours
- From [2]

From [1]

- MDVS
- Anonymous Ours
- MDRS-PKE
5. Our MDRS-PKE Scheme Construction

- **IK-CPA**
- **PKE (IND-CPA)**
- **Simulation Sound NIZK**
- **Statistically Binding Commitment Scheme**
- **Anonymous**
- **From [2]**
- **Ours**
- **From [1]**

MDRS-PKE

Anonymous

From [1]

ETH Zürich

Eurocrypt' 22 33/35
5. Our MDRS-PKE Scheme Construction

<table>
<thead>
<tr>
<th>PKEBC</th>
<th>Anonymous</th>
<th>From [2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKE (IND-CPA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NIZK</td>
<td>Simulation Sound</td>
<td></td>
</tr>
<tr>
<td>Statistically Binding</td>
<td>Commitment Scheme</td>
<td></td>
</tr>
<tr>
<td>MDRS-PKE</td>
<td>Anonymous</td>
<td>Ours</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From [1]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ours</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDVS</td>
</tr>
</tbody>
</table>

IK-CPA

From [2]

Ours

Anonymous

MDVS

From [1]

Ours

Anonymous

MDRS-PKE

Eurocrypt' 22 33/35
5. Our MDRS-PKE Scheme Construction

- NIZK Simulation Sound PKE (IND-CPA)
- Commitment Scheme Statistically Binding
- From [2]
- Anonymous
- From [1]
- Anonymous
- MDVS
- From [1]
- Anonymous
- MDRS-PKE
- From [1]
- Anonymous
- IDK-CPA
- PKE (IND-CPA)
5. Our MDRS-PKE Scheme Construction

- **IK-CPA**
  - PKE (IND-CPA)
  - From [2]
- **Simulation Sound**
  - NIZK
  - From [2]
- **Statistically Binding**
  - Commitment Scheme
- **From [1]**
  - Ours
  - Anonymous
- **From [1]**
  - Ours
  - Anonymous
  - MDVS
  - Ours
  - Anonymous
  - MDRS-PKE
  - From [1]

*Eurocrypt' 22*
5. Our MDRS-PKE Scheme Construction

- **PKE (IND-CPA)**: NIZK Simulation Sound PKE
- **Commitment Scheme**: Statistically Binding
- **IK-CPA Anonymous**: From [2]
- **Ours Anonymous**: From [1]
- **PKEBC Anonymous**: Ours From [2]
- **MDVS Anonymous**: Ours From [1]
Thank you!
Bibliography

   Stronger security and constructions of multi-designated verifier signatures.

   Public-key cryptosystems provably secure against chosen ciphertext attacks.

   Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security.
6. Full MDRS-PKE construction
6. MDRS-PKE Scheme Construction

MDRS-PKE.Setup:

\[
\begin{align*}
pp_{\text{MDVS}} &\leftarrow \text{MDVS.\textit{Setup}}(1^k); \\
pp_{\text{PKEBC}} &\leftarrow \text{PKEBC.\textit{Setup}}(1^k); \\
\text{Output } pp &= (pp_{\text{MDVS}}, pp_{\text{PKEBC}});
\end{align*}
\]

MDRS-PKE.Gen\textit{Snd}(pp = (pp_{\text{MDVS}}, pp_{\text{PKEBC}})):

\[
\begin{align*}
(\text{spk}_{\text{MDVS}}, \text{ssk}_{\text{MDVS}}) &\leftarrow \text{MDVS.\textit{Gen}}_{\text{Sig}}(pp_{\text{MDVS}}); \\
\text{Output } (\text{spk} := \text{spk}_{\text{MDVS}}, \text{ssk} := (\text{spk}, \text{ssk}_{\text{MDVS}}));
\end{align*}
\]

MDRS-PKE.Gen\textit{Rcv}(pp = (pp_{\text{MDVS}}, pp_{\text{PKEBC}})):

\[
\begin{align*}
(\text{vpk}_{\text{MDVS}}, \text{vsk}_{\text{MDVS}}) &\leftarrow \text{MDVS.\textit{Gen}}_{\text{V}}(pp_{\text{MDVS}}); \\
(\text{rpk}_{\text{PKEBC}}, \text{rsk}_{\text{PKEBC}}) &\leftarrow \text{PKEBC.\textit{Gen}}(pp_{\text{PKEBC}}); \\
\text{Output } \left(rpk := (\text{vpk}_{\text{MDVS}}, \text{rpk}_{\text{PKEBC}}), \text{rsk} := (\text{rpk}, (\text{vsk}_{\text{MDVS}}, \text{rsk}_{\text{PKEBC}}))\right);
\end{align*}
\]
6. MDRS-PKE Scheme Construction

MDRS-PKE.\textit{Enc}(ssk_i := (spk_i, ssk_{MDVS_i}), \vec{v} := (\text{rpk}_1, \ldots, \text{rpk}_{|\vec{v}|}), m): \\
\hspace{1cm}(\text{where } \text{rpk}_i := (\text{vpk}_{MDVS_i}, \text{rpk}_{PKEBC_i}))

Let \vec{v}_{PKEBC} = (\text{rpk}_{PKEBC_1}, \ldots, \text{rpk}_{PKEBC_{|\vec{v}|}});
Let \vec{v}_{MDVS} = (\text{vpk}_{MDVS_1}, \ldots, \text{vpk}_{MDVS_{|\vec{v}|}});
\sigma \leftarrow \text{MDVS.}\text{Sign}(ssk_{MDVS_i}, \vec{v}_{MDVS}, (\vec{v}_{PKEBC}, m));
\text{Output } \text{PKEBC.}\text{Enc}(\vec{v}_{PKEBC}, (spk_i, \vec{v}_{MDVS}, m, \sigma));
6. MDRS-PKE Scheme Construction

MDRS-PKE.\textit{Dec}(rsk_j := (rpk_j, (vsk_{MDVS_j}, rsk_{PKEBC_j})), c):
(\text{where } rpk_j := (vpk_{MDVS_j}, rpk_{PKEBC_j}))

\begin{align*}
\vec{v}_{PKEBC}, (spk_i, \vec{v}_{MDVS}, m, \sigma)) & \leftarrow \text{PKEBC.D}(rsk_{PKEBC_j}, c); \\
\text{Output } \bot & \text{ if } (\vec{v}_{PKEBC}, (spk_i, \vec{v}_{MDVS}, m, \sigma)) = \bot \text{ or } |\vec{v}_{PKEBC}| \neq |\vec{v}_{MDVS}|; \\
\text{Let } \vec{v} & = ((v_{MDVS_1}, v_{PKEBC_1}), \ldots, (v_{MDVS_{|\vec{v}_{PKEBC}|}}, v_{PKEBC_{|\vec{v}_{PKEBC}|}})); \\
\text{Output } \bot & \text{ if } rpk_j \not\in \vec{v} \text{ or } \text{MDVS.Vfy}(spk_i, vsk_{MDVS_j}, \vec{v}_{MDVS}, (\vec{v}_{PKEBC}, m), \sigma) \neq \text{valid}; \\
\text{Output } & (spk_i, \vec{v}, m); 
\end{align*}