Trapdoor functions from the Computational Diffie-Hellman Assumption

Sanjam Garg¹ Mohammad Hajiabadi^{1,2}

¹University of California, Berkeley ²University of Virginia

August 22, 2018

Classical Public-Key Crypto

TDF [DH76]





Classical Public-Key Crypto

TDF [DH76]



New Description in Crypting agety

Institute

And the Company of t

PKE [GM82]



Probabilistic Excreption & New To Pier Restal Polor Excepting Secret And Indonesia of the Control of the Contro

8. Introduction
The same proposes as Berrypsia Shiness
to passes the Montey propose
to passes the Montey appropria
to passes the Montey appropria
to passes the Montey appropria
to passes the same to expendent, onetion, the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to the same
to the same to the same to t

Profession for complete data to filence, measure for the filency is been found to Complete and Party and the interactional of Complete and the interactional of the interactional of some problems in number of the interactional control of their control of the interactional contr

Pagarante la processo de la compansa del compansa de la compansa de la compansa del compansa de la compansa del la compansa del la compansa de la compansa del la comp

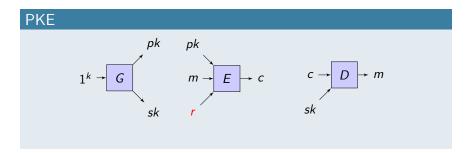
We could fill be a point out the basis excitation and the basis of the

could be invasid.

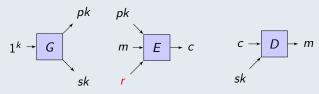
Though it on the little in a break the IRRA or
the Rathin exhause, in team of these information fashin exhause, in team of these information,
in present that describing is based without any
attention fash, in his unleases, derending in lored his
attents that, in his unleases, derending in lored his
attention fash, in his unleases, derending in lored his
attention fash of the lored of the lored in the lored in
the control of the control of the lored in the lored in
the control of Trappines Functions in
the lored in the lored

replaced by Probabilistic Energytims. Inmorphy each message we make set of a formonth The encoding of each message will figured on the message plus the result of a requestive of roots toware. Consequently, there are morphy possible exceedings for each message. However, messages are always employ decodable.

Medicine Severation is company, defense two most by the contract of the contract of the medicine Severation is company; defense two most to the contract of the contract of the medicine Severation is company; defense two most to the contract of the contract of the medicine Severation is company; defense two most to the contract of the medicine Severation is considered to the contract of the medicine Severation is considered to the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contract of the contract of the severation of the contract of the contr

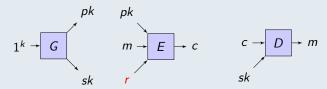






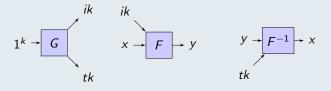
Security: $\forall m_0, m_1 : (pk, E(pk, m_0; r_0)) \stackrel{c}{=} (pk, E(pk, m_1; r_1))$

PKE

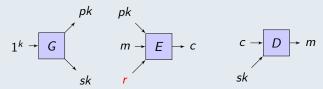


Security: $\forall m_0, m_1 : (pk, E(pk, m_0; r_0)) \stackrel{c}{=} (pk, E(pk, m_1; r_1))$

TDF

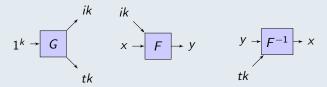


PKE



Security: $\forall m_0, m_1 : (pk, E(pk, m_0; r_0)) \stackrel{c}{=} (pk, E(pk, m_1; r_1))$

TDF



One-wayness Security: $(ik, F(ik, x)) \stackrel{?}{\rightarrow} x$ is hard for random ik, x.

TDF vs PKE

Main Difference

▶ No randomness used in the evaluation algorithm of TDF.

TDF vs PKE

Main Difference

▶ No randomness used in the evaluation algorithm of TDF.

Relations

► TDF implies the existence of PKE. [Yao'82, GM'82].

TDF vs PKE

Main Difference

▶ No randomness used in the evaluation algorithm of TDF.

Relations

- ► TDF implies the existence of PKE. [Yao'82, GM'82].
- ► TDF impossible from PKE w.r.t. black-box techniques [GMR'01].





 ik_1 , ik_2 and tk_1



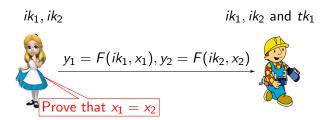
 ik_1, ik_2

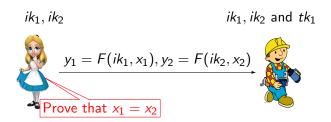
 ik_1 , ik_2 and tk_1



$$y_1 = F(ik_1, x_1), y_2 = F(ik_2, x_2)$$

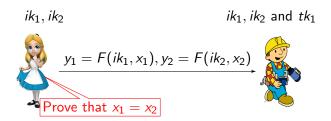






Bob: Compute $x_1 = F^{-1}(tk_1, y_1)$ and check if $y_2 = F(ik_2, x_1)$.

► Application: black-box constructions of CCA-secure PKE ([PW'08,RS'09, etc]).



Bob: Compute $x_1 = F^{-1}(tk_1, y_1)$ and check if $y_2 = F(ik_2, x_1)$.

► Application: black-box constructions of CCA-secure PKE ([PW'08,RS'09, etc]).

PKE instead of TDF

► Consistency check: require some kind of proof (e.g., NIZK). [BFY90,NY90]

What assumptions are sufficient for TDFs?

- ► Factoring
- ► DDH and LWE [PW08]

What assumptions are sufficient for TDFs?

- ► Factoring
- ► DDH and LWE [PW08]

Big gap from PKE!

What assumptions are sufficient for TDFs?

- Factoring
- ▶ DDH and LWE [PW08]

Big gap from PKE!

This talk: We can do it from CDH.

CDH and **DDH**

 \mathbb{G} : group of order p and generator g.

CDH and **DDH**

 \mathbb{G} : group of order p and generator g.

Computational Diffie-Hellman (CDH)

▶ Hard to compute g^{xy} from (g, g^x, g^y) , where $x, y \leftarrow \mathbb{Z}_p$.

CDH and **DDH**

 \mathbb{G} : group of order p and generator g.

Computational Diffie-Hellman (CDH)

▶ Hard to compute g^{xy} from (g, g^x, g^y) , where $x, y \leftarrow \mathbb{Z}_p$.

Decisional Diffie-Hellman (DDH)

• $(g, g^x, g^y, g^{xy}) \stackrel{c}{=} (g, g^x, g^y, g^z)$, where $x, y, z \leftarrow \mathbb{Z}_p$

Why is CDH Preferable?

Why is CDH Preferable?

- CDH is a weaker assumption.
 - ► There are groups in which CDH is conjectured to be hard but DDH is easy (e.g., \mathbb{Z}_p^* , groups with pairings).

Main Challenge in Building TDF from DH-Related Assumptions

Why is constructing TDF from Diffie-Hellman assumptions difficult?

Main Challenge in Building TDF from DH-Related Assumptions

Why is constructing TDF from Diffie-Hellman assumptions difficult? It doesn't naturally offer trapdoors!

TDF from DDH (Failed Idea Using ElGamal Encryption)

TDF from DDH (Failed Idea Using ElGamal Encryption)

$$(\mathbb{G}, g), |\mathbb{G}| = p.$$

$$pk = g^{\alpha} \quad pk \quad c = (g^r, pk^r \cdot m)$$

$$sk = \alpha \quad r \quad sk = \alpha \quad m$$

TDF from DDH (Failed Idea Using ElGamal Encryption)

$$(\mathbb{G}, g), |\mathbb{G}| = p.$$

$$pk = g^{\alpha} \quad pk \quad c = (g^r, pk^r \cdot m)$$

$$sk = \alpha \quad r \quad sk = \alpha \quad m$$

Main bottleneck in designing TDFs

▶ Recovering *r*: solving the Discrete Log!

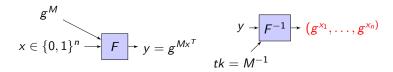
$$(\mathbb{G},g)$$
, $|\mathbb{G}|=p$.

$$(\mathbb{G},g)$$
, $|\mathbb{G}|=p$.

▶ $ik = g^M$ where $M \in \mathbb{Z}_p^{n \times n}$ (and invertible) and $tk = M^{-1}$

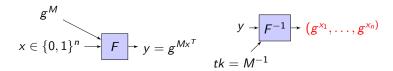
$$(\mathbb{G},g)$$
, $|\mathbb{G}|=p$.

▶ $ik = g^M$ where $M \in \mathbb{Z}_p^{n \times n}$ (and invertible) and $tk = M^{-1}$



$$(\mathbb{G},g)$$
, $|\mathbb{G}|=p$.

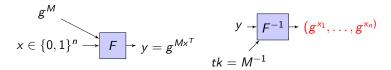
▶ $ik = g^M$ where $M \in \mathbb{Z}_p^{n \times n}$ (and invertible) and $tk = M^{-1}$



► Can solve discrete-log as $x_1 ... x_n \in \{0, 1\}!$

$$(\mathbb{G},g)$$
, $|\mathbb{G}|=p$.

▶ $ik = g^M$ where $M \in \mathbb{Z}_p^{n \times n}$ (and invertible) and $tk = M^{-1}$



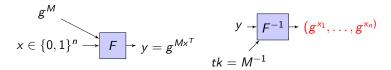
▶ Can solve discrete-log as $x_1 ... x_n \in \{0, 1\}!$

One-wayness

Matrix pseudorandomness [NR97]: DDH implies $g^M \stackrel{c}{\equiv} g^{M'}$, where M is a random invertible matrix and M' is a random rank-one matrix.

$$(\mathbb{G},g)$$
, $|\mathbb{G}|=p$.

▶ $ik = g^M$ where $M \in \mathbb{Z}_p^{n \times n}$ (and invertible) and $tk = M^{-1}$



▶ Can solve discrete-log as $x_1 ... x_n \in \{0, 1\}!$

One-wayness

- Matrix pseudorandomness [NR97]: DDH implies $g^M \stackrel{c}{\equiv} g^{M'}$, where M is a random invertible matrix and M' is a random rank-one matrix.
- CDH is not known to imply rank indistinguishability.

- 1 Background
 - Introduction
 - Main Challenges
- 2 Our TDF Construction
 - Our Methodology
 - Base Primitive: Recyclable Targeting KEM
 - TDF from Recyclable Targeting KEM
- 3 Summary and Future Work

Our Methodology for building TDF from CDH

► Derandomizing a class of PKE

Our Methodology for building TDF from CDH

- ▶ Derandomizing a class of PKE
 - ► TDFs from recyclable targeted key-encapsulation schemes (Recyclable Targeted KEMs) [DG'17, BBS'03]

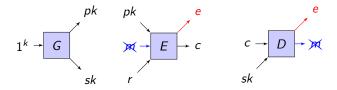
Our Methodology for building TDF from CDH

- ► Derandomizing a class of PKE
 - ► TDFs from recyclable targeted key-encapsulation schemes (Recyclable Targeted KEMs) [DG'17, BBS'03]

Plan for the Rest of the talk

- ▶ Define Recyclable Targeted KEM
- ► CDH ⇒ Recyclable Targeted KEM (Not discussed. See [DG'17].)
- ▶ Recyclable Targeted KEM ⇒ TDF

Key-Encapsulation Mechanism



e is always a single bit.

Targeting Property [DG'17]

- ► E(pk, (i, b); r) = (ct, e)
- ▶ D(sk, ct) = e if $(pk, sk) \in K(1^{\lambda})$ and $sk_i = b$.

Targeting Property [DG'17]

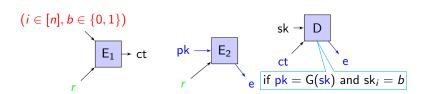
- ightharpoonup $\mathsf{E}(\mathsf{pk}, (i, b); r) = (\mathsf{ct}, e)$
- ▶ D(sk, ct) = e if $(pk, sk) \in K(1^{\lambda})$ and $sk_i = b$.
- ► Security: $(pk, sk, ct, e) \stackrel{c}{=} (pk, sk, ct, e')$, where $(ct, e) \stackrel{\$}{\leftarrow} E(pk, (i, 1 sk_i); r)$ and $e' \stackrel{\$}{\leftarrow} \{0, 1\}$.

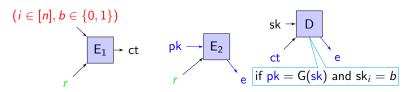
Targeting Property [DG'17]

- ightharpoonup E(pk, (i, b); r) = (ct, e)
- ▶ D(sk, ct) = e if $(pk, sk) \in K(1^{\lambda})$ and $sk_i = b$.
- ► Security: $(pk, sk, ct, e) \stackrel{c}{=} (pk, sk, ct, e')$, where $(ct, e) \stackrel{\$}{\leftarrow} E(pk, (i, 1 sk_i); r)$ and $e' \stackrel{\$}{\leftarrow} \{0, 1\}$.

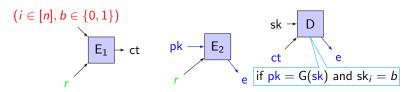
Recyclability

ct does not depend on pk. So $E(pk, (i, b); r) = (E_1((i, b); r), E_2(pk, (i, b); r)) = (ct, e)$



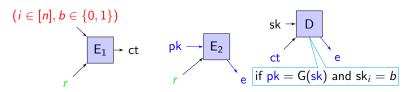


$$\blacktriangleright \mathsf{tk} = \begin{pmatrix} r_1 \\ r_1' \end{pmatrix} \mathsf{and} \mathsf{ik} = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_1' \end{pmatrix} = \begin{pmatrix} \mathsf{E}_1((i=1,b=0);r_1) \\ \mathsf{E}_1((i=1,b=1);r_1') \end{pmatrix}$$



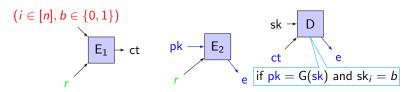
▶ tk =
$$\begin{pmatrix} r_1 \\ r'_1 \end{pmatrix}$$
 and ik = $\begin{pmatrix} \operatorname{ct}_1 \\ \operatorname{ct}'_1 \end{pmatrix}$ = $\begin{pmatrix} \operatorname{E}_1((i=1,b=0);r_1) \\ \operatorname{E}_1((i=1,b=1);r'_1) \end{pmatrix}$

► F(ik, sk):



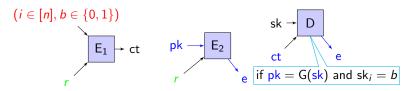
▶ tk =
$$\begin{pmatrix} r_1 \\ r'_1 \end{pmatrix}$$
 and ik = $\begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}'_1 \end{pmatrix}$ = $\begin{pmatrix} \mathsf{E}_1((i=1,b=0);r_1) \\ \mathsf{E}_1((i=1,b=1);r'_1) \end{pmatrix}$

F(ik, sk): let pk = G(sk).



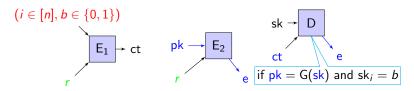
▶ tk =
$$\begin{pmatrix} r_1 \\ r'_1 \end{pmatrix}$$
 and ik = $\begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}'_1 \end{pmatrix}$ = $\begin{pmatrix} \mathsf{E}_1((i=1,b=0);r_1) \\ \mathsf{E}_1((i=1,b=1);r'_1) \end{pmatrix}$

- ► F(ik, sk): let pk = G(sk).
 - if $sk_1 = 0$, then return $(pk, D(sk, ct_1))$



$$\blacktriangleright \mathsf{tk} = \begin{pmatrix} r_1 \\ r_1' \end{pmatrix} \mathsf{and} \mathsf{ik} = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_1' \end{pmatrix} = \begin{pmatrix} \mathsf{E}_1((i=1,b=0);r_1) \\ \mathsf{E}_1((i=1,b=1);r_1') \end{pmatrix}$$

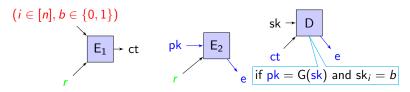
- ► F(ik, sk): let pk = G(sk).
 - if $sk_1 = 0$, then return $(pk, D(sk, ct_1)) = (pk, E_2(pk; r_1))$.
 - if $sk_1 = 1$, then return $(pk, D(sk, ct'_1)) = (pk, E_2(pk; r'_1))$.



$$\blacktriangleright \mathsf{tk} = \begin{pmatrix} r_1 \\ r_1' \end{pmatrix} \mathsf{and} \mathsf{ik} = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_1' \end{pmatrix} = \begin{pmatrix} \mathsf{E}_1((i=1,b=0);r_1) \\ \mathsf{E}_1((i=1,b=1);r_1') \end{pmatrix}$$

- ► F(ik, sk): let pk = G(sk).
 - if $sk_1 = 0$, then return $(pk, D(sk, ct_1)) = (pk, E_2(pk; r_1))$.
 - if $sk_1 = 1$, then return $(pk, D(sk, ct'_1)) = (pk, E_2(pk; r'_1))$.
- ▶ F^{-1} : Check for a match:

$$\begin{pmatrix} \mathsf{E}_2(\mathsf{pk}; r_1) \\ \mathsf{E}_2(\mathsf{pk}; r_1') \end{pmatrix}$$

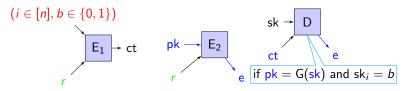


$$\blacktriangleright \mathsf{tk} = \begin{pmatrix} r_1 \\ r_1' \end{pmatrix} \mathsf{and} \mathsf{ik} = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_1' \end{pmatrix} = \begin{pmatrix} \mathsf{E}_1((i=1,b=0);r_1) \\ \mathsf{E}_1((i=1,b=1);r_1') \end{pmatrix}$$

- ► F(ik, sk): let pk = G(sk).
 - if $sk_1 = 0$, then return $(pk, D(sk, ct_1)) = (pk, E_2(pk; r_1))$.
 - if $sk_1 = 1$, then return $(pk, D(sk, ct'_1)) = (pk, E_2(pk; r'_1))$.
- ightharpoonup F⁻¹: Check for a match:

$$\begin{pmatrix} \mathsf{E}_2(\mathsf{pk}; r_1) \\ \mathsf{E}_2(\mathsf{pk}; r_1') \end{pmatrix}$$

► Can recover sk_1 with probability 1/2. This can be boosted via repetition.

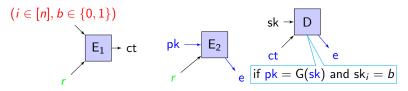


$$\blacktriangleright \mathsf{tk} = \begin{pmatrix} r_1 \\ r_1' \end{pmatrix} \mathsf{and} \mathsf{ik} = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_1' \end{pmatrix} = \begin{pmatrix} \mathsf{E}_1((i=1,b=0);r_1) \\ \mathsf{E}_1((i=1,b=1);r_1') \end{pmatrix}$$

- ► F(ik, sk): let pk = G(sk).
 - if $sk_1 = 0$, then return $(pk, D(sk, ct_1)) = (pk, E_2(pk; r_1))$.
 - if $sk_1 = 1$, then return $(pk, D(sk, ct'_1)) = (pk, E_2(pk; r'_1))$.
- \triangleright F⁻¹: Check for a match:

$$\begin{pmatrix} \mathsf{E}_2(\mathsf{pk}; r_1) \\ \mathsf{E}_2(\mathsf{pk}; r_1') \end{pmatrix}$$

- ► Can recover sk_1 with probability 1/2. This can be boosted via repetition.
- ► Not clear how to prove security!



$$\blacktriangleright \mathsf{tk} = \begin{pmatrix} r_1 \\ r_1' \end{pmatrix} \mathsf{and} \mathsf{ik} = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_1' \end{pmatrix} = \begin{pmatrix} \mathsf{E}_1((i=1,b=0);r_1) \\ \mathsf{E}_1((i=1,b=1);r_1') \end{pmatrix}$$

- ► F(ik, sk): let pk = G(sk).
 - if $sk_1 = 0$, then return $(pk, D(sk, ct_1)) = (pk, E_2(pk; r_1))$.
 - if $sk_1 = 1$, then return $(pk, D(sk, ct'_1)) = (pk, E_2(pk; r'_1))$.
- \triangleright F⁻¹: Check for a match:

$$\begin{pmatrix} \mathsf{E}_2(\mathsf{pk}; r_1) \\ \mathsf{E}_2(\mathsf{pk}; r_1') \end{pmatrix}$$

- ► Can recover sk_1 with probability 1/2. This can be boosted via repetition.
- ► Not clear how to prove security!
 - ▶ Fix: Put a random bit in the place you cannot apply *D*.

Recovering the First Bit

▶ Gen(1 $^{\lambda}$): tk = $\binom{r_1}{r'_1}$ and

$$\mathsf{ik} = \left(\begin{smallmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_1' \end{smallmatrix} \right) = \left(\begin{smallmatrix} \mathsf{E_1}((i = 1, b = 0); r_1) \\ \mathsf{E_1}((i = 1, b = 1); r_1') \end{smallmatrix} \right)$$

- ▶ $F(ik, sk||b_1)$: let pk = G(sk). Then:
 - $\begin{tabular}{l} \blacktriangleright & \mbox{if } \mathsf{sk}_1 = 0 \mbox{ then } \mathsf{M}_1 := \begin{pmatrix} \mathsf{D}(\mathsf{sk},\mathsf{ct}_1) \\ \mathsf{b}_1 \\ \end{tabular} \begin{tabular}{l} \mathsf{b}_1 \\ \mathsf{b}_1 \\ \mbox{bissing} \begin{tabular}{l} \mathsf{b}_1 \\ \mathsf{D}(\mathsf{sk},\mathsf{ct}_1') \\ \end{tabular} \begin{tabular}{l} \mathsf{b}_1 \\ \mbox{D}(\mathsf{sk},\mathsf{ct}_1') \\ \mbox{D}(\mathsf{sk},\mathsf{ct}$

Return $Y = (pk, M_1)$.

► F⁻¹(tk, Y):

$$\mathsf{M}_1' = \begin{pmatrix} \mathsf{E}_2(\mathsf{pk}, (1, 0); r_1) \\ \mathsf{E}_2(\mathsf{pk}, (1, 1); r_1') \end{pmatrix}$$

Summary and Future Work

Summary

► A Construction of TDFs from CDH.

Future Work

- ► Extended forms of TDFs from CDH (e.g., lossy trapdoor functions)
- ► Trapdoor Permutations from CDH/DDH?