Efficient Privacy-Preserving Biometric Identification

Yan Huang Lior Malka David Evans Jonathan Katz

University of Virginia · University of Maryland · University of Virginia · University of Maryland

http://www.mightbeevil.org/secure-biometrics/

Feb 9, 2011

Motivating Scenario: Private No-Fly Checking

Any Matches?

Yes/No

Threat Models

- *Semi-honest* adversary
  - Must follow the protocol correctly
- *Malicious* adversary
  - Can deviate arbitrarily from the protocol

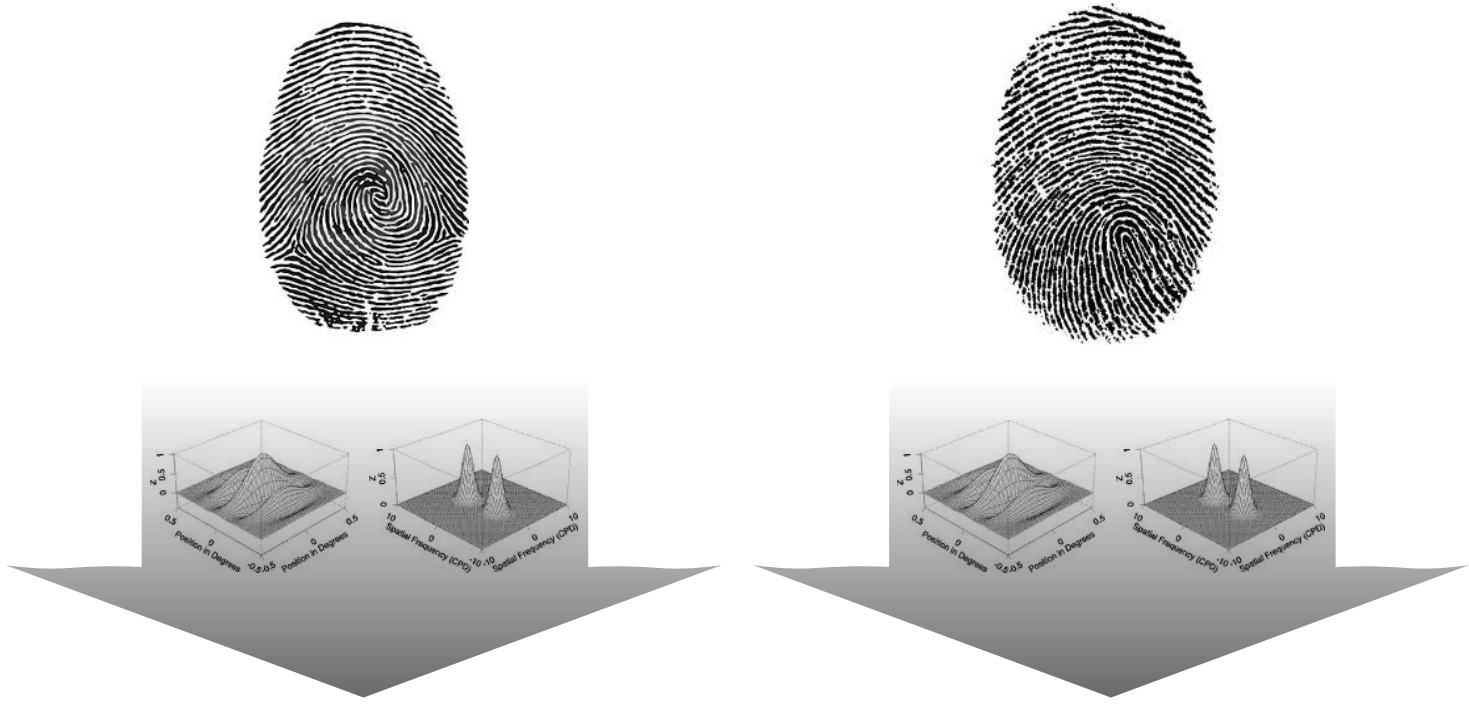
In both threat models, an adversary attempts to break either the *correctness* or the *privacy* property of the protocol.

Threat Models

- ***Semi-honest* adversary**
  - **Must follow the protocol correctly**
- *Malicious* adversary
  - Can deviate arbitrarily from the protocol

In both threat models, an adversary attempts to break either the *correctness* or the *privacy* property of the protocol.

Filterbank-based Fingerprint Recognition [Jain et al., 2000]



Also used by Barni et al. [2010].

Non-private Protocol

$V = \{v_1, v_2, \cdots, v_M\}$

$v' = \{v'_1, v'_2, \cdots, v'_N\}$

Euclidean Distance

$d = \{d_1, d_2, \cdots, d_M\}$

Finding Minimum

$d^* = \min_{1 \le i \le M} (d_i)$

Retrieve Identity

Record(i^*), if $d^* = d_{i^*} < \varepsilon$;
$\perp$, otherwise.

Privacy-preserving Protocol

$V = \{v_1, v_2, \cdots, v_M\}$

$v' = \{v'_1, v'_2, \cdots, v'_N\}$

Euclidean Distance — **Homomorphic Encryption**

$d = \{d_1, d_2, \cdots, d_M\}$

Finding Minimum — **Garbled Circuits**

$d^* = \min_{1 \le i \le M} (d_i)$

Retrieve Identity — **Backtracking Protocol**

Record(i^*), if $d^* = d_{i^*} < \varepsilon$;
$\perp$, otherwise.

Privacy-preserving Protocol

$V = \{v_1, v_2, \cdots, v_M\}$

$v' = \{v'_1, v'_2, \cdots, v'_N\}$

Euclidean Distance

Homomorphic Encryption

$d = \{d_1, d_2, \cdots, d_M\}$

Finding Minimum

$d^* = \min_{1 \le i \le M} (d_i)$

Retrieve Identity

Record(i^*), if $d^* = d_{i^*} < \varepsilon$;
$\perp$, otherwise.

Euclidean Distance

Let d_i be the distance between $\boldsymbol{v}_i = [v_{i,j}]_{1 \le j \le N}$ and $\boldsymbol{v}' = [v'_j]_{1 \le j \le N}$

$$d_i = \|\boldsymbol{v}_i - \boldsymbol{v}'\|^2 = \sum_{j=1}^{N} (v_{i,j} - v'_j)^2$$

$$= \underbrace{\sum_{j=1}^{N} v_{i,j}^2}_{S_{i,1}} + \underbrace{\sum_{j=1}^{N} (-2 v_{i,j} \cdot v'_j)}_{S_{i,2}} + \underbrace{\sum_{j=1}^{N} {v'_j}^2}_{S_3}$$

For privacy, want to compute $[\![d_i]\!]_{\mathsf{pk}}$.

Additive Homomorphic Encryption

$$\left.\begin{array}{l} [\![a]\!]_{\mathsf{pk}} \\ \\ [\![b]\!]_{\mathsf{pk}} \end{array}\right\} \Longrightarrow [\![a+b \mod p]\!]_{\mathsf{pk}} = [\![a]\!]_{\mathsf{pk}} \cdot [\![b]\!]_{\mathsf{pk}}$$

$$\left.\begin{array}{l} [\![a]\!]_{\mathsf{pk}} \\ \\ c \end{array}\right\} \Longrightarrow [\![c \cdot a \mod p]\!]_{\mathsf{pk}} = [\![a]\!]^{c}_{\mathsf{pk}}$$

We used Paillier cryptosystem [Catalano et al., 2001, Paillier, 1999] in our prototype.

Additive Homomorphic Encryption

$$\left.\begin{array}{l}[\![a]\!] \\ \\ [\![b]\!]\end{array}\right\} \Longrightarrow [\![a+b \mod p]\!] = [\![a]\!] \cdot [\![b]\!]$$

$$\left.\begin{array}{l}[\![a]\!] \\ \\ c\end{array}\right\} \Longrightarrow [\![c \cdot a \mod p]\!] = [\![a]\!]^c$$

We used Paillier cryptosystem [Catalano et al., 2001, Paillier, 1999] in our prototype.

Private Euclidean Distance

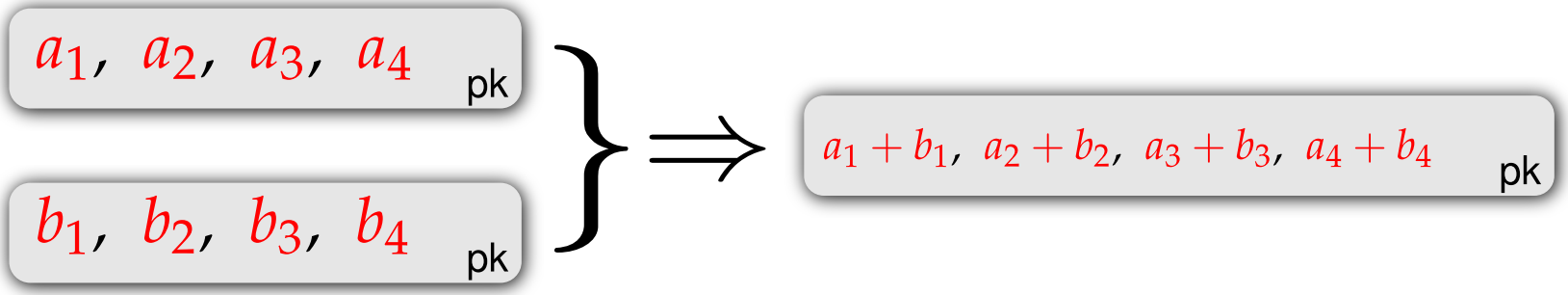
$$[\![d_i]\!] = \Big[\!\Big[\underbrace{\sum_{j=1}^{N} v_{i,j}^2}_{S_{i,1}} + \underbrace{\sum_{j=1}^{N} (-2 v_{i,j} v'_j)}_{S_{i,2}} + \underbrace{\sum_{j=1}^{N} {v'_j}^2}_{S_3} \Big]\!\Big]$$

$$= [\![S_{i,1}]\!] \cdot [\![S_{i,2}]\!] \cdot [\![S_3]\!]$$

$$[\![S_{i,2}]\!] = \left[\!\!\left[\sum_{j=1}^{N} (-2 v_{i,j} v'_j) \right]\!\!\right] = \prod_{j=1}^{N} [\![-2 v_{i,j}]\!]^{v'_j}$$

Improving the Efficiency

- **Modular exponentiation is slow.** For every i, computing $[\![S_{i,2}]\!]$ requires N modular exponentiations. Overall, it involves MN modular exponentiations
- **Encode many messages in one homomorphic encryption**



Packing was introduced by Sadeghi et al. [2009] to save bandwidth, but is exploited more aggressively here to save computation also.

Padding 0's to Ensure Correctness

$$
\begin{array}{rcl}
51,28,72 & & \boxed{51,28,72}_{pk} \\
+\ 39,92,22 & \bullet & \boxed{39,92,22}_{pk} \\
\hline
91,20,94 & & \boxed{91,20,94}_{pk}
\end{array}
$$

$$
\begin{array}{rcl}
051,028,072 & & \boxed{051,028,072}_{pk} \\
+\ 039,092,022 & \bullet & \boxed{039,092,022}_{pk} \\
\hline
090,120,094 & & \boxed{090,120,094}_{pk}
\end{array}
$$

Vertical Partitioning to Speedup Computing $[\![S_{i,2}]\!]$

$$[\![S_{i,2}]\!] = \prod_{j=1}^{N} [\![-2v_{i,j}]\!]^{v'_j}$$

$$\begin{bmatrix} -2v_{1,1} & -2v_{1,2} & \cdots & -2v_{1,N} \\ -2v_{2,1} & -2v_{2,2} & \cdots & -2v_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ -2v_{\kappa,1} & -2v_{\kappa,2} & \cdots & -2v_{\kappa,N} \end{bmatrix}$$

Vertical Partitioning to Speedup Computing $[\![S_{i,2}]\!]$

$$[\![S_{i,2}]\!] = \prod_{j=1}^{N} [\![-2v_{i,j}]\!]^{v'_j}$$

$$[\![S_{1,2} \| S_{2,2} \| \cdots \| S_{\kappa,2}]\!] = \prod_{1 \le j \le N} \left[\!\!\left[-2v_{1,j}v'_j \| -2v_{2,j}v'_j \| \cdots \| -2v_{\kappa,j}v'_j \right]\!\!\right]$$

$$\begin{bmatrix} -2v_{1,1} & -2v_{1,2} & \cdots & -2v_{1,N} \\ -2v_{2,1} & -2v_{2,2} & \cdots & -2v_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ -2v_{\kappa,1} & -2v_{\kappa,2} & \cdots & -2v_{\kappa,N} \end{bmatrix}$$

Vertical Partitioning to Speedup Computing $[\![S_{i,2}]\!]$

$$[\![S_{i,2}]\!] = \prod_{j=1}^{N} [\![-2v_{i,j}]\!]^{v'_j}$$

$$[\![S_{1,2} \| S_{2,2} \| \cdots \| S_{\kappa,2}]\!] = \prod_{1 \le j \le N} \left[\!\!\left[-2v_{1,j}v'_j \| -2v_{2,j}v'_j \| \cdots \| -2v_{\kappa,j}v'_j \right]\!\!\right]$$

$$\left[\!\!\left[-2v_{1,j}v'_j \| -2v_{2,j}v'_j \| \cdots \| -2v_{\kappa,j}v'_j \right]\!\!\right] = [\![-2v_{1,j} \| -2v_{2,j} \| \cdots \| -2v_{\kappa,j}]\!]^{v'_j}$$

$$\begin{bmatrix} -2v_{1,1} & -2v_{1,2} & \cdots & -2v_{1,N} \\ -2v_{2,1} & -2v_{2,2} & \cdots & -2v_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ -2v_{\kappa,1} & -2v_{\kappa,2} & \cdots & -2v_{\kappa,N} \end{bmatrix}$$

Vertical Partitioning to Speedup Computing $[\![S_{i,2}]\!]$

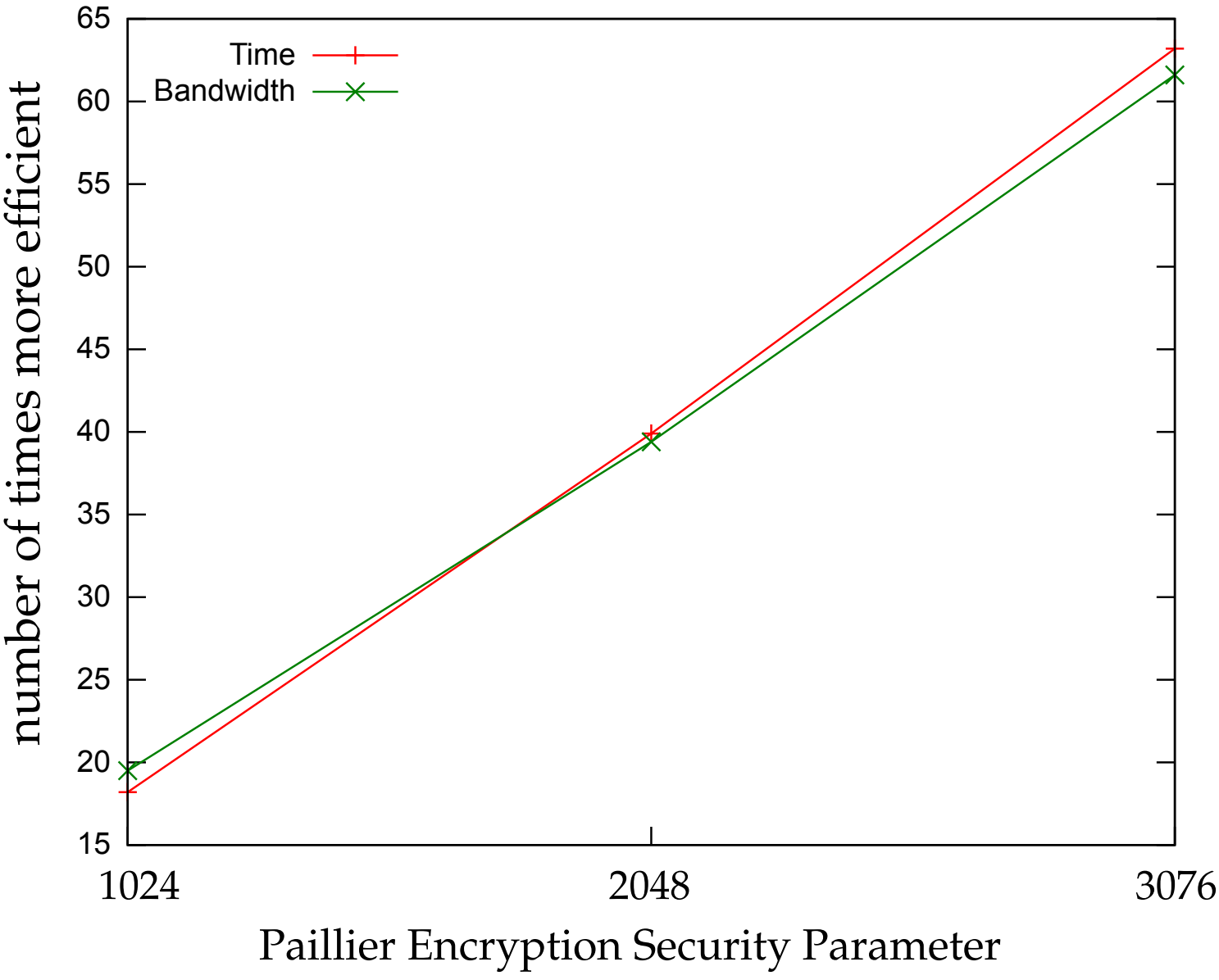
$$[\![S_{i,2}]\!] = \prod_{j=1}^{N} [\![-2v_{i,j}]\!]^{v'_j}$$

$$[\![S_{1,2} \| S_{2,2} \| \cdots \| S_{\kappa,2}]\!] = \prod_{1 \le j \le N} \left[\!\!\left[-2v_{1,j}v'_j \| -2v_{2,j}v'_j \| \cdots \| -2v_{\kappa,j}v'_j \right]\!\!\right]$$

$$\left[\!\!\left[-2v_{1,j}v'_j \| -2v_{2,j}v'_j \| \cdots \| -2v_{\kappa,j}v'_j \right]\!\!\right] = [\![-2v_{1,j} \| -2v_{2,j} \| \cdots \| -2v_{\kappa,j}]\!]^{v'_j}$$

$$\begin{bmatrix} -2v_{1,1} & -2v_{1,2} & \cdots & -2v_{1,N} \\ -2v_{2,1} & -2v_{2,2} & \cdots & -2v_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ -2v_{\kappa,1} & -2v_{\kappa,2} & \cdots & -2v_{\kappa,N} \end{bmatrix}$$

Effects of Packing



Sharing the Secrets

The server generates nonce masks $\boldsymbol{r} = [r_1, r_2, \cdots, r_M]$ and sends

$$[\![d'_1 \| \cdots \| d'_M]\!]_{\mathsf{pk}} = [\![(d_1 + r_1) \| (d_2 + r_2) \| \cdots \| (d_M + r_M)]\!]_{\mathsf{pk}}$$

where pk is the client's public key.

Server	**Client**
d'_1	r_1
d'_2	r_2
$\vdots$	$\vdots$
d'_M	r_M

Make the sampling range of r_i large enough so that d'_i and d_i is statistically indistinguishable.

Privacy-preserving Protocol

$V = \{v_1, v_2, \cdots, v_M\}$

$v' = \{v'_1, v'_2, \cdots, v'_N\}$

Euclidean Distance

$d' = \{d'_1, d'_2, \cdots, d'_M\}$

$r = \{r_1, r_2, \cdots, r_M\}$

Finding Minimum

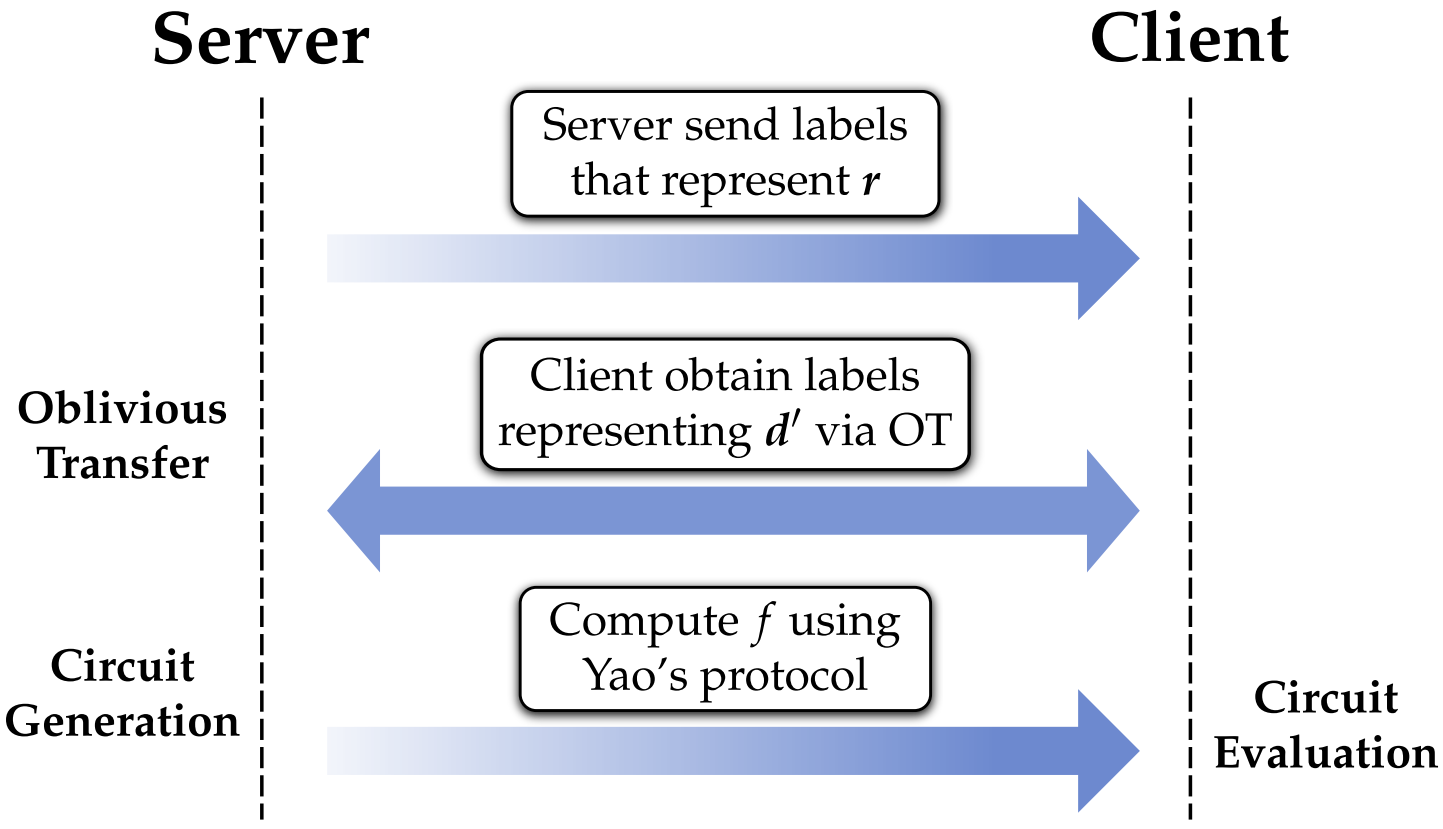
Garbled Circuits

$d^* = \min_{1 \le i \le M} (d_i)$

Retrieve Identity

Record(i^*), if $d^* = d_{i^*} < \varepsilon$;
$\perp$, otherwise.

Garbled Circuits Protocol

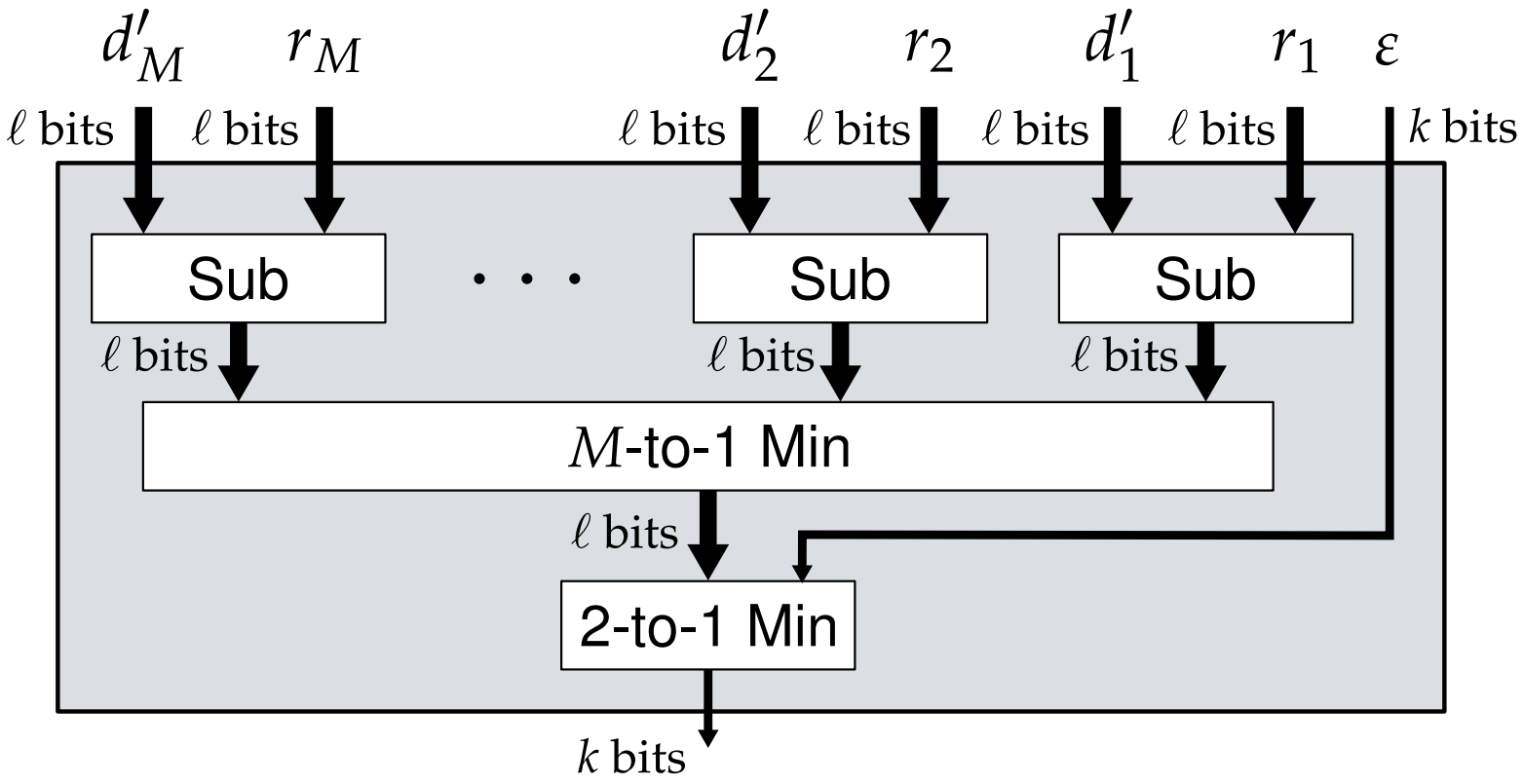


- Efficient oblivious transfer protocol combining schemes from both [Naor and Pinkas, 2001] and [Ishai et al., 2003]
- Standard garbled circuits [Yao, 1986] combined with free-XOR technique [Kolesnikov and Schneider, 2008]

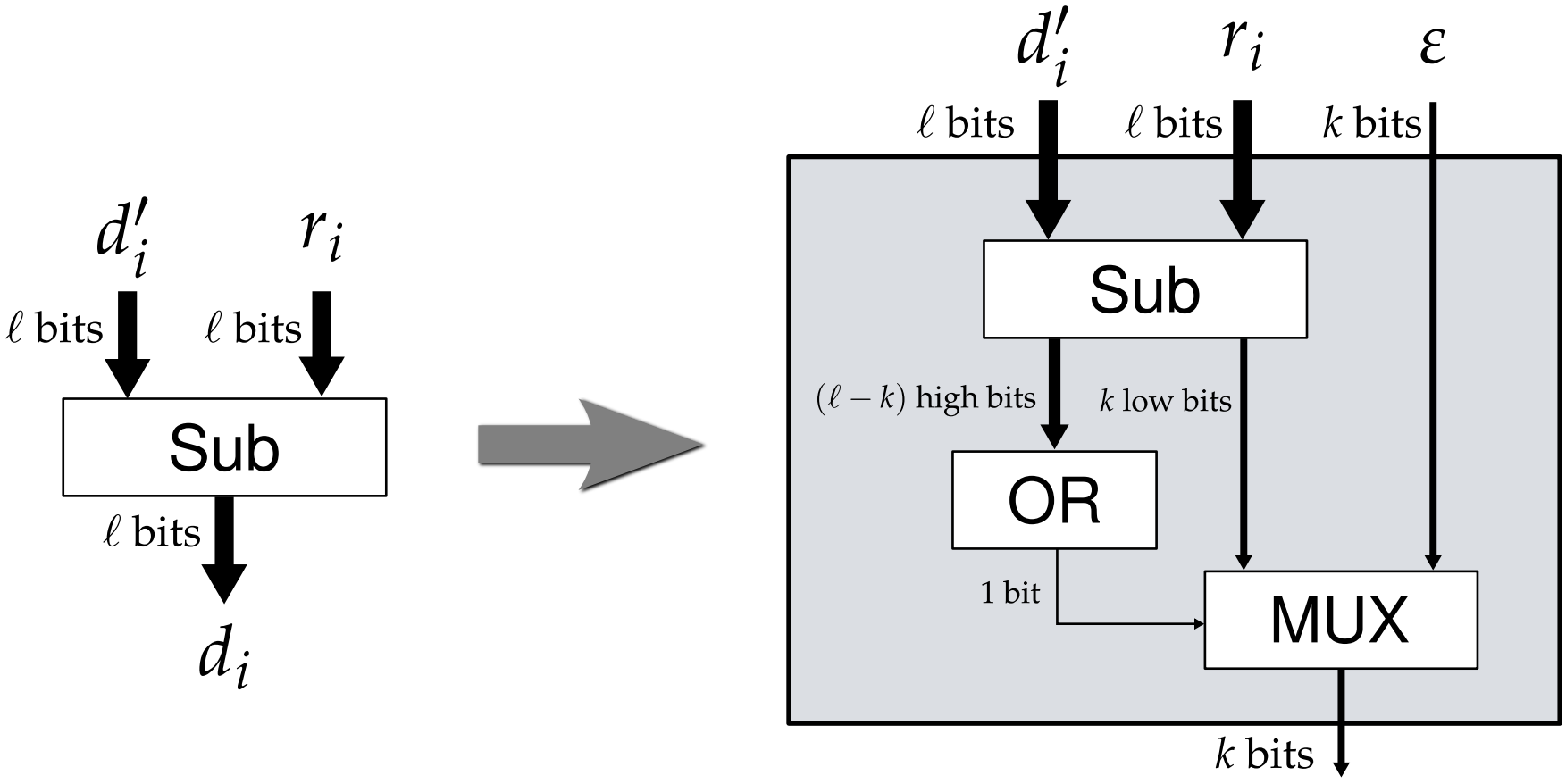
Finding the Minimum Differnce

Goal

Given $\boldsymbol{d'} = \boldsymbol{d} + \boldsymbol{r}$ and $\boldsymbol{r}$, securely compute $d^* = \min_{1 \le i \le M} (d_i, \varepsilon)$.



Reducing the Bit-width



Saves $2M(\ell - k)$ non-free gates in total.

Privacy-preserving Protocol

$V = \{v_1, v_2, \cdots, v_M\}$

$v' = \{v'_1, v'_2, \cdots, v'_N\}$

Euclidean Distance

Finding Minimum

$d^* = \min_{1 \le i \le M} (d_i)$

Retrieve Identity

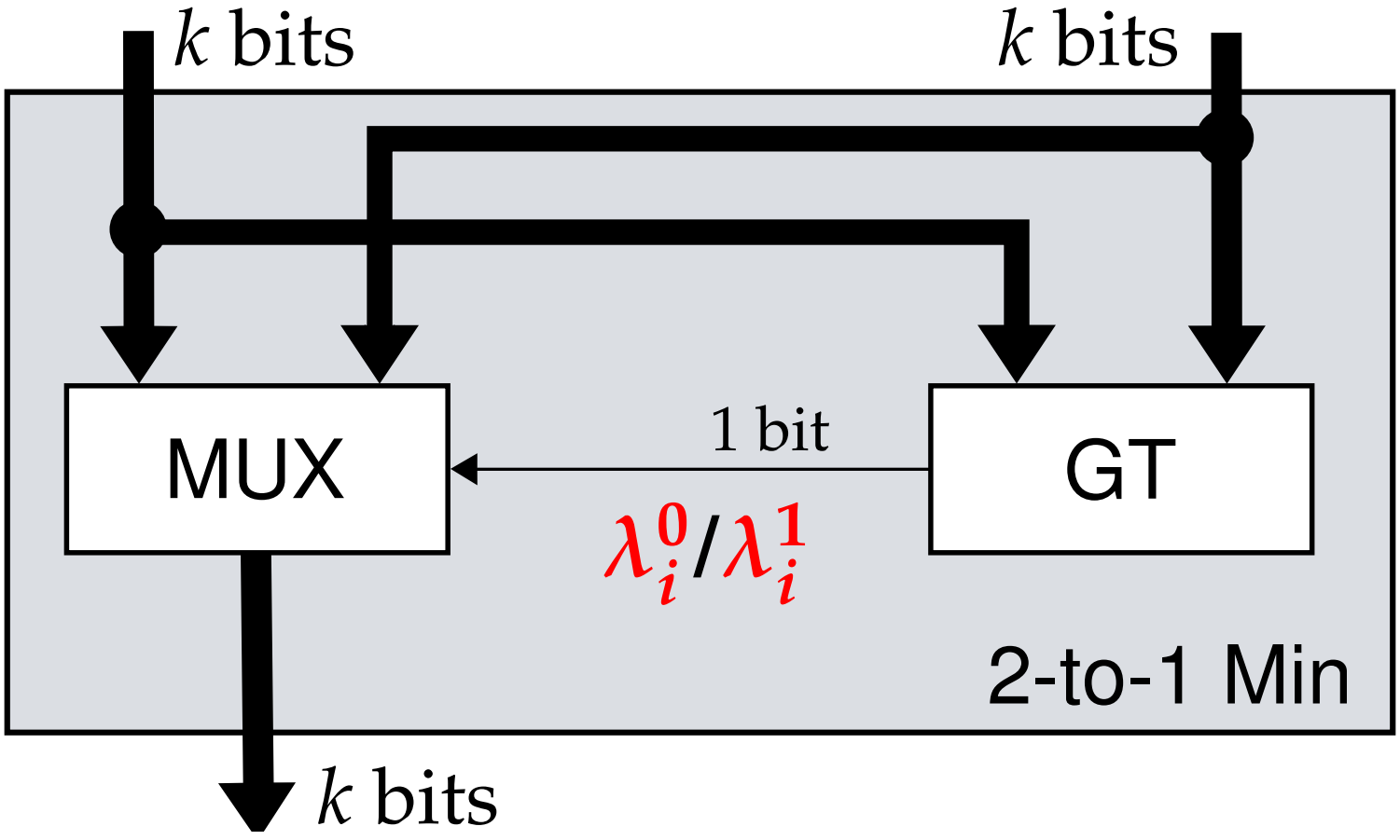
Backtracking Protocol

Record(i^*), if $d^* = d_{i^*} < \varepsilon$;
$\perp$, otherwise.

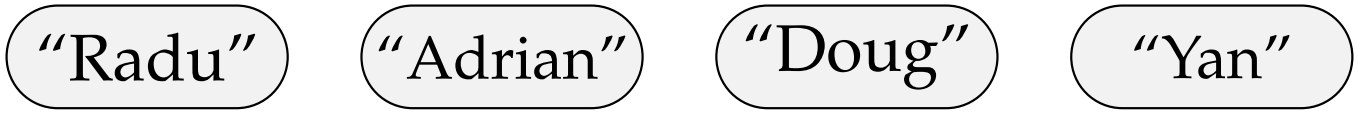
Finding the Record

- Ultimate goal is to retrieve the record associated with d^*
- Prior work [Kolesnikov et al., 2009] accomplished this by relaying indices throughout the M-to-1 Min circuit
- We achieve this with a *backtracking* protocol
  1. No need to propagate ID numbers
  2. Obtain record without an extra secure information retrieval by ID
  3. Use labels obtained in garbled circuit execution

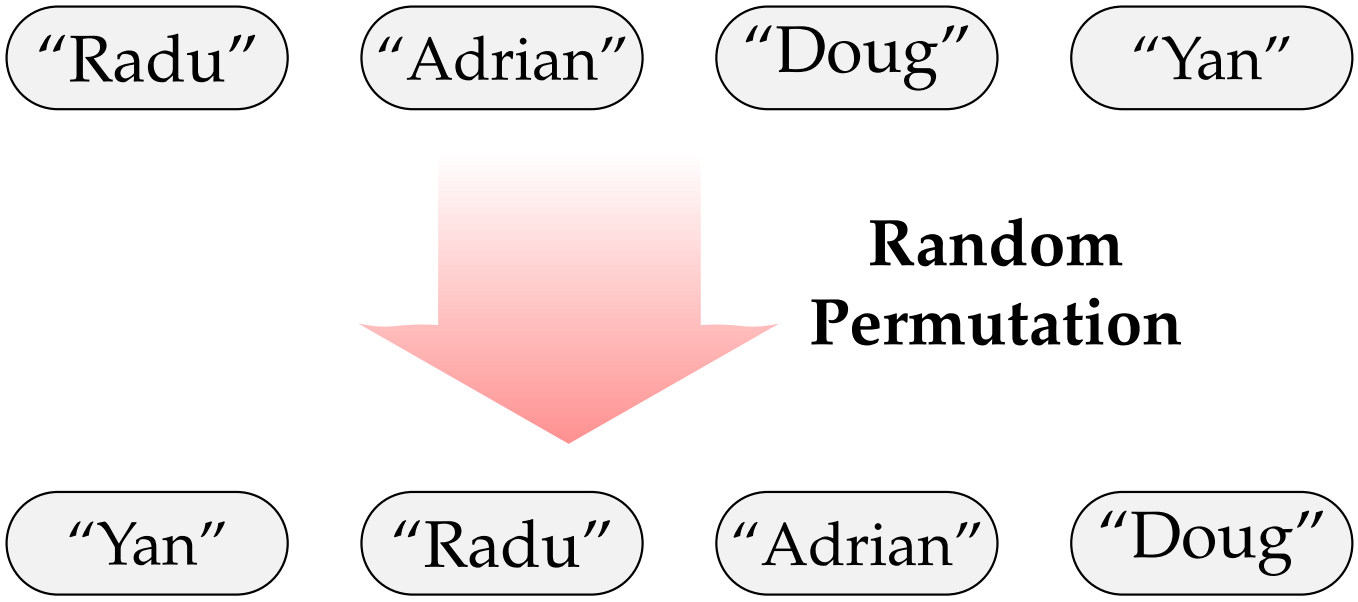
The 2-to-1 Min



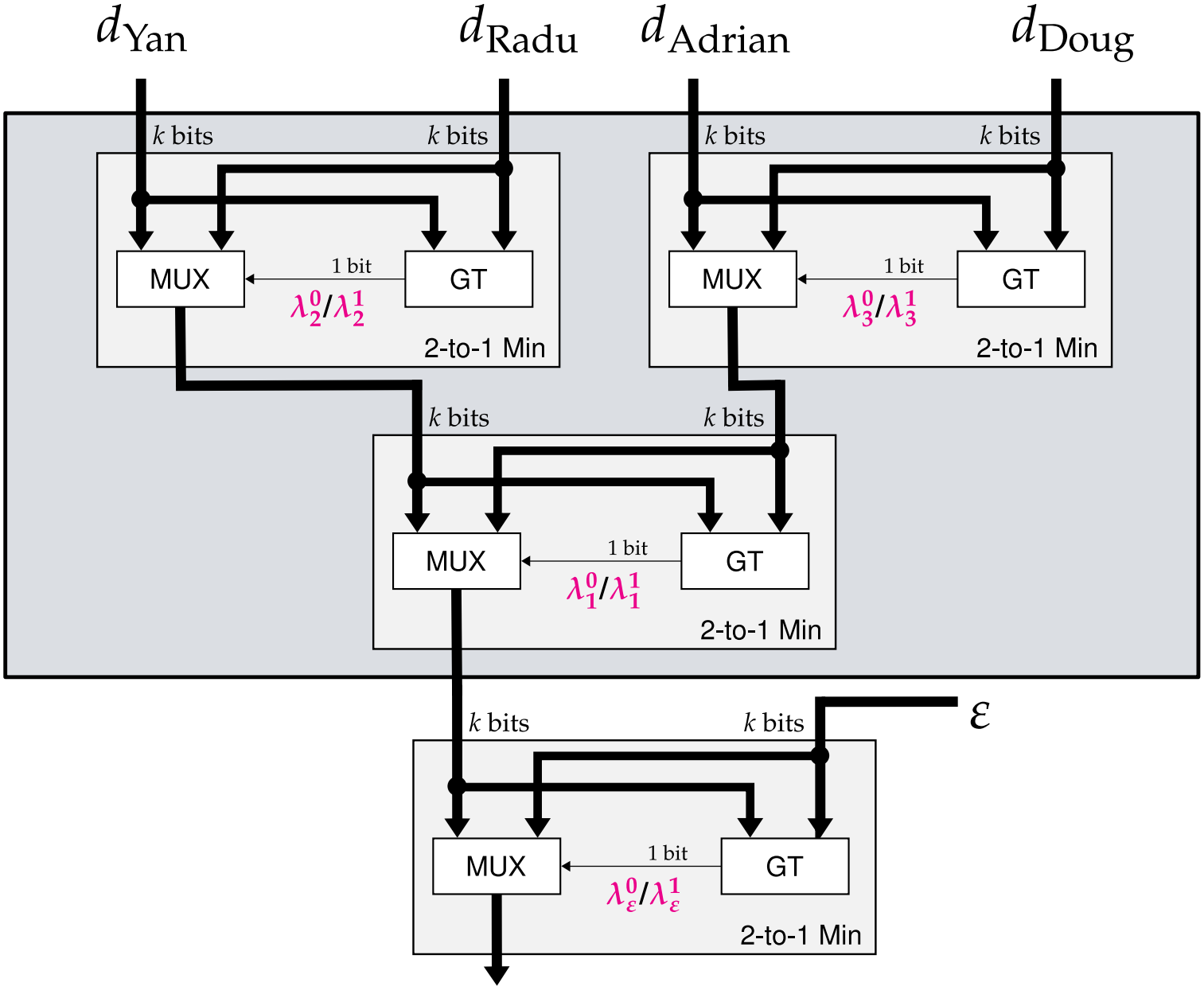
Mini Example — The Server



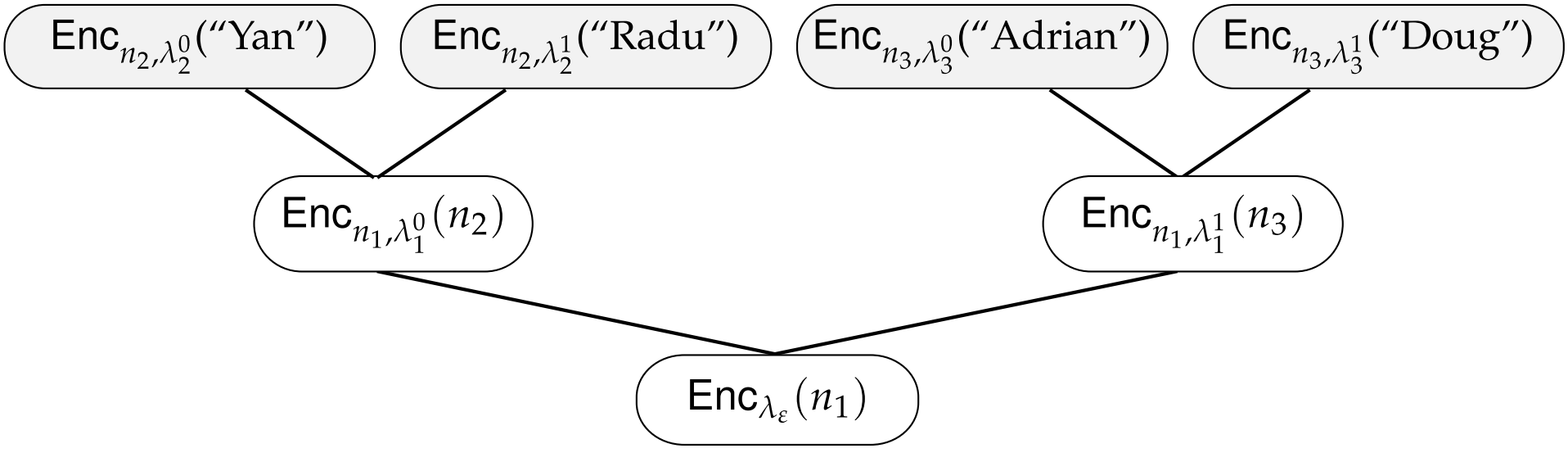
Mini Example — The Server



Selection Wires in the M-to-1 Min Tree

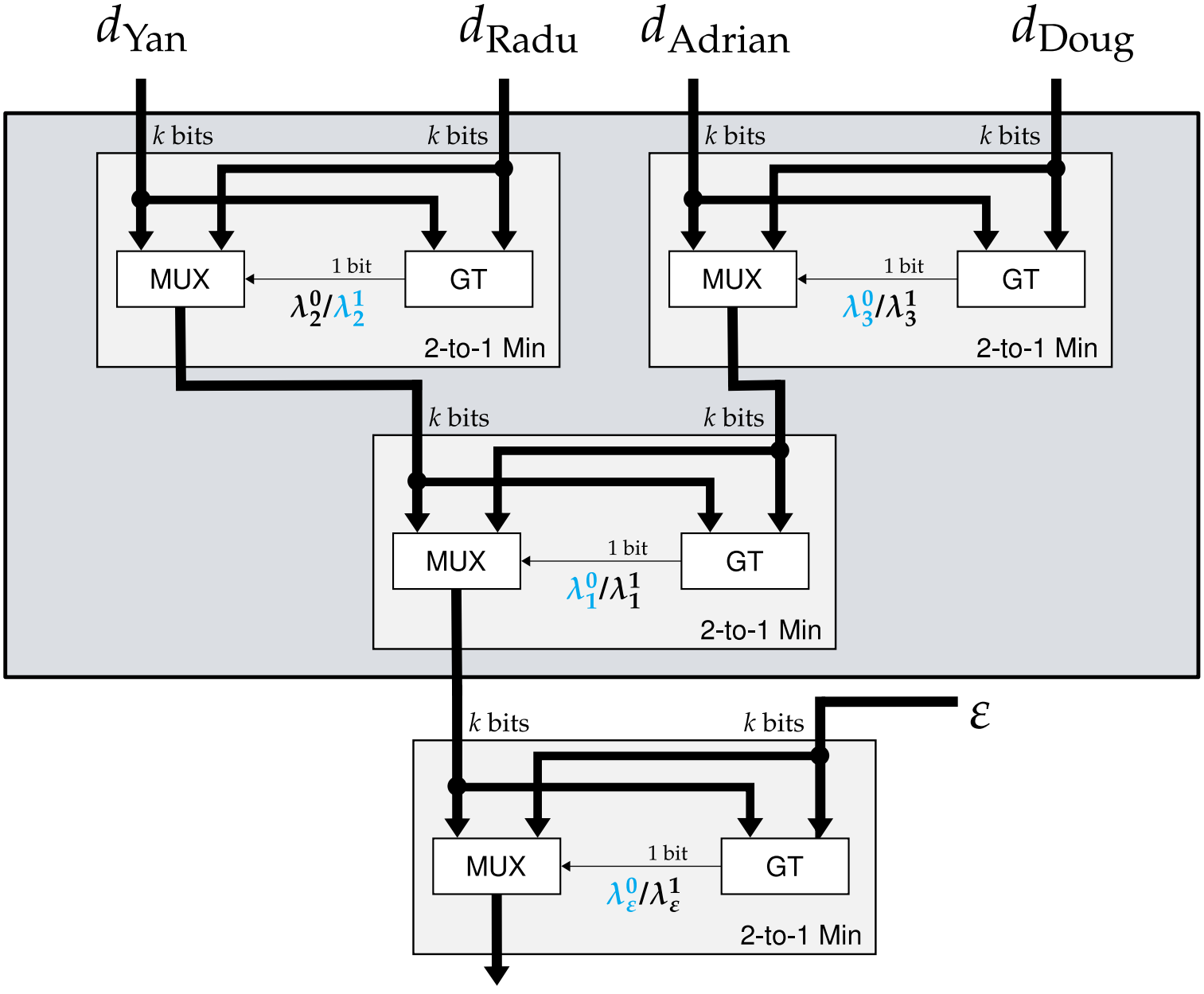


Backtracking — The Sender

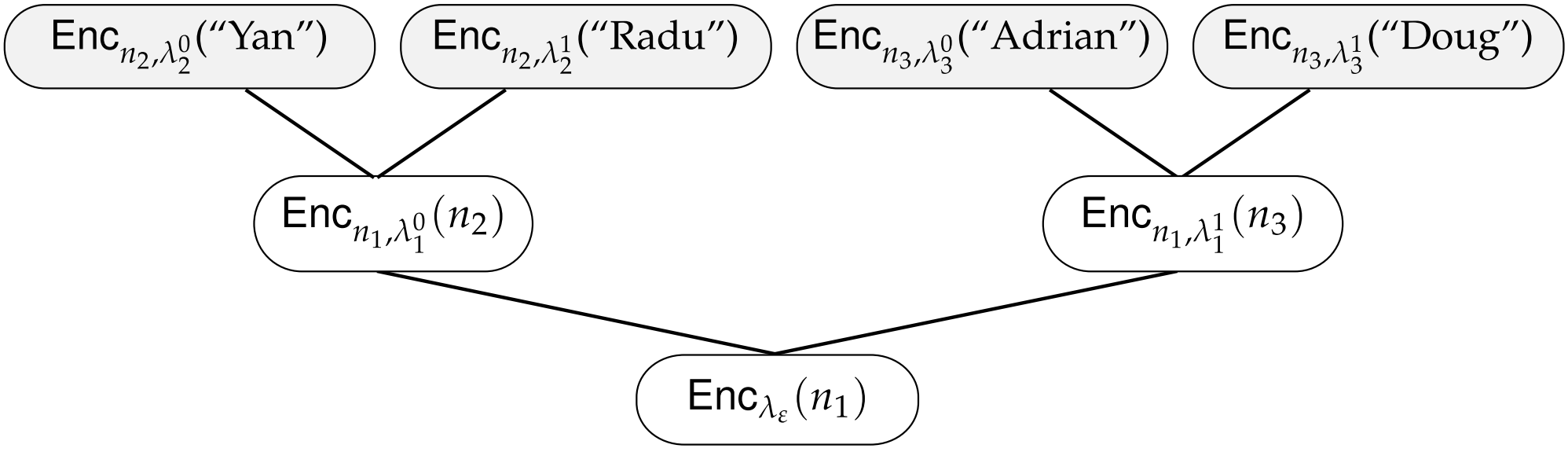


n_1, n_2, n_3 are random nonces known only to the sender.

Backtracking — The Receiver

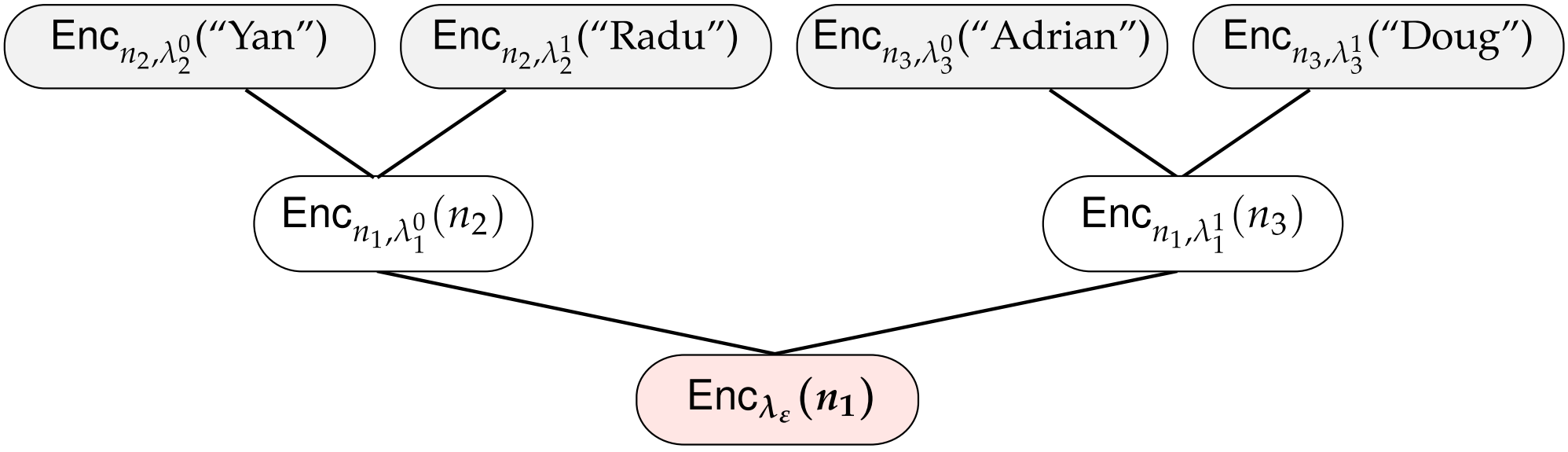


Backtracking — The Receiver



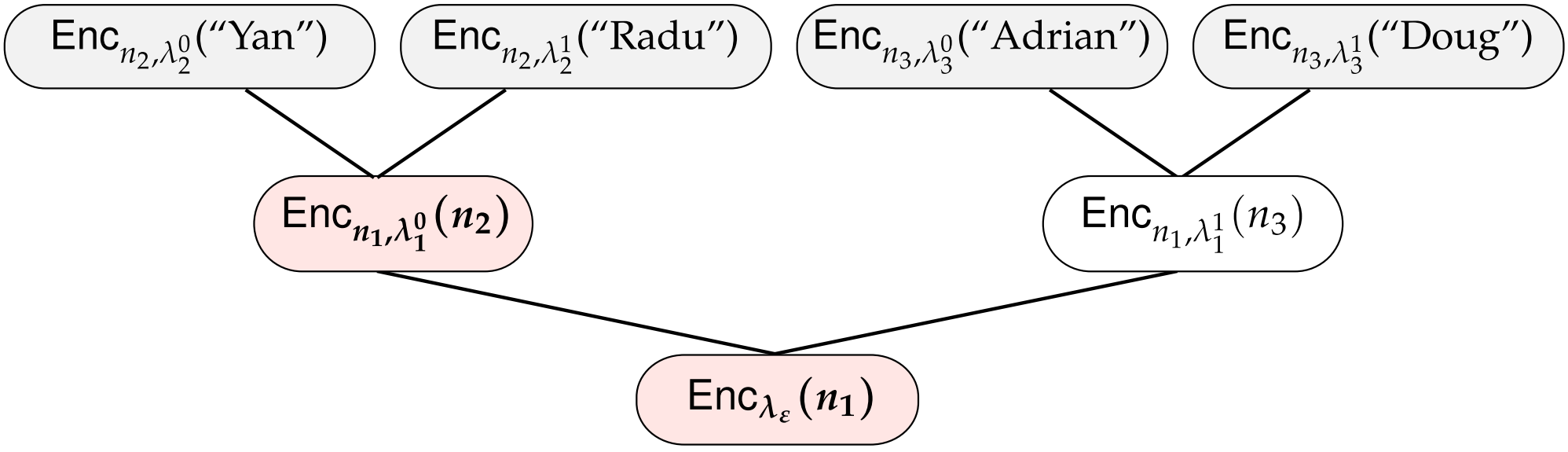
Client knows $\lambda_\varepsilon^0, \lambda_1^0, \lambda_2^1, \lambda_3^0$ from circuit evaluation,

Backtracking — The Receiver



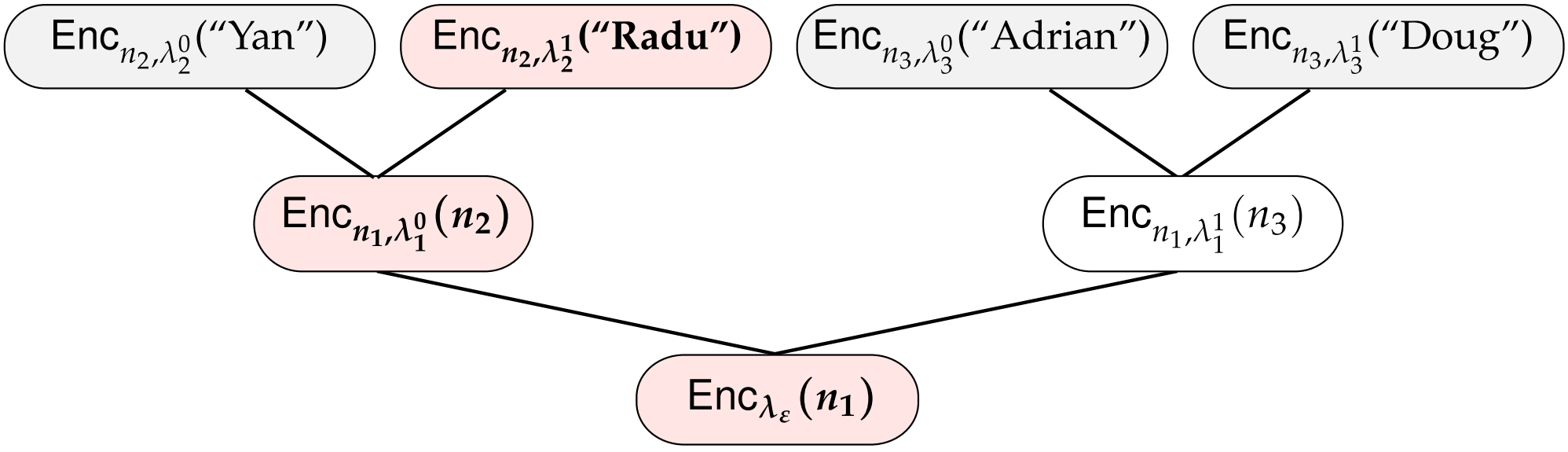
Client knows $\lambda_\varepsilon^0, \lambda_1^0, \lambda_2^1, \lambda_3^0$ from circuit evaluation, so is able to infer n_1

Backtracking — The Receiver



Client knows $\lambda_\varepsilon^0, \lambda_1^0, \lambda_2^1, \lambda_3^0$ from circuit evaluation, so is able to infer n_1, n_2

Backtracking — The Receiver



Client knows $\lambda_\varepsilon^0, \lambda_1^0, \lambda_2^1, \lambda_3^0$ from circuit evaluation, so is able to infer n_1, n_2, and **Radu**.

System Recap

$V = \{v_1, v_2, \cdots, v_M\}$

$v' = \{v'_1, v'_2, \cdots, v'_N\}$

Euclidean Distance — Distance

$d = \{d_1, d_2, \cdots, d_M\}$

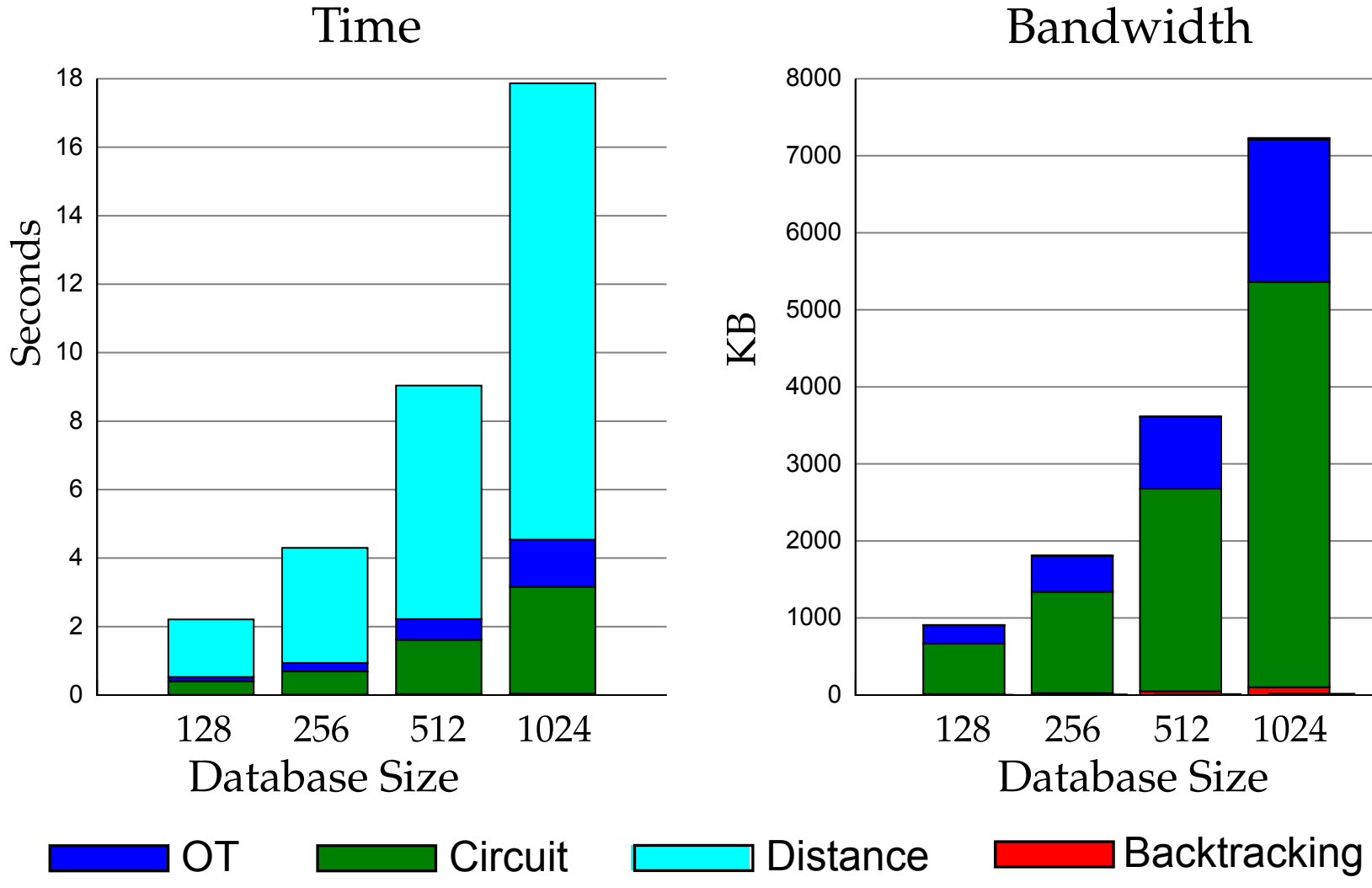
Finding Minimum — OT Circuit

$d^* = \min_{1 \le i \le M} (d_i)$

Retrieve Identity — Backtracking

Record(i^*), if $d^* = d_{i^*} < \varepsilon$;
$\perp$, otherwise.

Results — Online Performance



4.6× faster and uses 58% less bandwidth than Barni et al. [2010], even though we compute the global minimum

Thank you!

Software available for download at:

http://www.mightbeevil.org/secure-biometrics/

References I

Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Faillia, D. Fiore, R. Lazzeretti, V. Piuri, F. Scotti, and A. Piva. Privacy-Preserving Fingercode Authentication. In *ACM Multimedia and Security Workshop*, 2010.

Dario Catalano, Rosario Gennaro, Nick Howgrave-Graham, and Phong Nguyen. Paillier's Cryptosystem Revisited. In *ACM Conference on Computer and Communications Security*, 2001.

Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. Extending Oblivious Transfers Efficiently. In *CRYPTO*, 2003.

Anil Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti. Filterbank-based Fingerprint Matching. *IEEE Transactions on Image Processing*, pages 846–859, January 2000.

Vladimir Kolesnikov and Thomas Schneider. Improved Garbled Circuit: Free XOR Gates and Applications. In *International Colloquium on Automata, Languages and Programming*, 2008.

Vladimir Kolesnikov, Ahmad-Reza Sadeghi, and Thomas Schneider. Improved Garbled Circuit Building Blocks and Applications to Auctions and Computing Minima. In *International Conference on Cryptology and Network Security*, 2009.

Moni Naor and Benny Pinkas. Efficient Oblivious Transfer Protocols. In *ACM-SIAM Symposium on Discrete Algorithms*, 2001.

Pascal Paillier. Public-key Cryptosystems based on Composite Degree Residuosity Classes. *EUROCRYPT*, 1999.

Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. Efficient Privacy-Preserving Face Recognition. In *International Conference on Information Security and Cryptology*, 2009.

Andrew Yao. How to Generate and Exchange Secrets. In *Symposium on Foundations of Computer Science*, 1986.