# Cryptographically secure multiparty evaluation of system reliability

Louis J. M. Aslett (louis.aslett@durham.ac.uk)

Department of Mathematical Sciences
Durham University

1st UK Reliability Meeting
3 April 2019

UTOPIÆ Uncertainty Treatment and Optimisation in Aerospace Engineering

MARIE CURIE ACTIONS

Durham University

# Introduction

# General Motivation

Security in statistical applications is a growing concern:

- computing in a ‘hostile’ environment (e.g. cloud computing);

# General Motivation

Security in statistical applications is a growing concern:

- computing in a ‘hostile’ environment (e.g. cloud computing);
- donation of sensitive/personal data (e.g. medical/genetic studies);

# General Motivation

Security in statistical applications is a growing concern:

- computing in a ‘hostile’ environment (e.g. cloud computing);
- donation of sensitive/personal data (e.g. medical/genetic studies);
- complex models on constrained devices (e.g. smart watches)

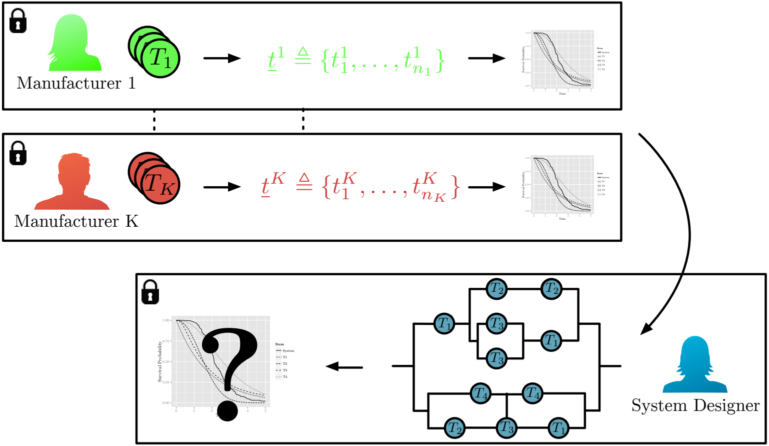
# General Motivation

Security in statistical applications is a growing concern:

- computing in a 'hostile' environment (e.g. cloud computing);
- donation of sensitive/personal data (e.g. medical/genetic studies);
- complex models on constrained devices (e.g. smart watches)
- running confidential algorithms on confidential data (e.g. engineering reliability — topic of this talk)

# Motivation in Reliability Theory

Inference on system/network reliability whilst *maintaining privacy requirements* of all parties.



# Homomorphic Encryption

# Encryption the solution?

Encryption can provide security guarantees …

Easy

$\texttt{Enc}(k_p, m) \rightleftharpoons c$ $\qquad\qquad$ $\texttt{Dec}(k_s, c) = m$

Hard without $k_s$

… but is typically 'brittle'.

# Encryption the solution?

Encryption can provide security guarantees …

Easy

$$\texttt{Enc}(k_p, m) \rightleftharpoons c \qquad\qquad \texttt{Dec}(k_s, c) = m$$

Hard without $k_s$

… but is typically 'brittle'.

Rivest et al. (1978) proposed encryption schemes capable of arbitrary addition and multiplication may be possible. Gentry (2009) showed first **fully homomorphic encryption** scheme.

# Encryption the solution?

Encryption can provide security guarantees …

Easy

$$\texttt{Enc}(k_p, m) \rightleftharpoons c \qquad \texttt{Dec}(k_s, c) = m$$

Hard without $k_s$

… but is typically 'brittle'.

Rivest et al. (1978) proposed encryption schemes capable of arbitrary addition and multiplication may be possible. Gentry (2009) showed first **fully homomorphic encryption** scheme.

$$m_1 \qquad m_2 \xrightarrow{+} m_1 + m_2$$

# Encryption the solution?

Encryption can provide security guarantees ...

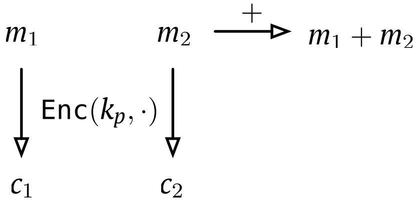
Easy

$$\texttt{Enc}(k_p, m) \rightleftharpoons c \qquad \texttt{Dec}(k_s, c) = m$$

Hard without $k_s$

... but is typically 'brittle'.

Rivest et al. (1978) proposed encryption schemes capable of arbitrary addition and multiplication may be possible. Gentry (2009) showed first **fully homomorphic encryption** scheme.



# Encryption the solution?

Encryption can provide security guarantees ...

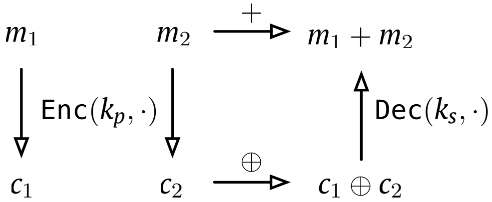
Easy

$$\texttt{Enc}(k_p, m) \rightleftharpoons c \qquad \texttt{Dec}(k_s, c) = m$$

Hard without $k_s$

... but is typically 'brittle'.

Rivest et al. (1978) proposed encryption schemes capable of arbitrary addition and multiplication may be possible. Gentry (2009) showed first **fully homomorphic encryption** scheme.



# Encryption the solution?

Encryption can provide security guarantees …

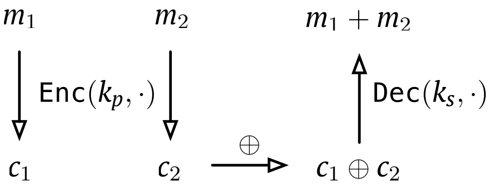
Easy

$$\texttt{Enc}(k_p, m) \rightleftharpoons c \qquad\qquad \texttt{Dec}(k_s, c) = m$$

Hard without $k_s$

… but is typically ‘brittle’.

Rivest et al. (1978) proposed encryption schemes capable of arbitrary addition and multiplication may be possible. Gentry (2009) showed first **fully homomorphic encryption** scheme.



# Limitations of homomorphic encryption

1. Message space (what we can encrypt)
    - Commonly only easy to encrypt binary/integers/polynomials
2. Cipher text size (the result of encryption)
    - Present schemes all inflate the size of data substantially (e.g. 1MB $\to$ 16.4GB)
3. Computational cost (computing without decrypting)
    - 1000's additions per sec
    - $\approx$ 50 multiplications per sec
4. Division and comparison operations (equality/inequality checks)
    - Not possible in current schemes!
5. Depth of operations
    - After a certain depth of multiplications, need to 'refresh' cipher text: hugely time consuming, so avoid!

# Encrypted Reliability Theory

# System lifetimes

Let $C_t^k \in \{0, 1, \ldots, M_k\}$ be random variable denoting number of components of type $k$ surviving at time $t$. Then, survival function of system lifetime $T_S$ is:

$$\begin{aligned}
\mathbb{P}(T_S > t) &= \sum_{l_1=0}^{M_1} \cdots \sum_{l_K=0}^{M_K} \Phi(l_1, \ldots, l_K)\, \mathbb{P}\left( \bigcap_{k=1}^{K} \{C_t^k = l_k\} \right) \\
&= \sum_{l_1=0}^{M_1} \cdots \sum_{l_K=0}^{M_K} \Phi(l_1, \ldots, l_K) \prod_{k=1}^{K} \mathbb{P}\left( C_t^k = l_k \right)
\end{aligned}$$

if the component types are independent.

# System lifetimes

Let $C_t^k \in \{0, 1, \dots, M_k\}$ be random variable denoting number of components of type $k$ surviving at time $t$. Then, survival function of system lifetime $T_S$ is:

$$\begin{aligned}\mathbb{P}(T_S > t) &= \sum_{l_1=0}^{M_1} \cdots \sum_{l_K=0}^{M_K} \Phi(l_1, \dots, l_K)\, \mathbb{P}\left( \bigcap_{k=1}^{K} \{C_t^k = l_k\} \right) \\ &= \sum_{l_1=0}^{M_1} \cdots \sum_{l_K=0}^{M_K} \Phi(l_1, \dots, l_K) \prod_{k=1}^{K} \mathbb{P}\left( C_t^k = l_k \right)\end{aligned}$$

if the component types are independent.

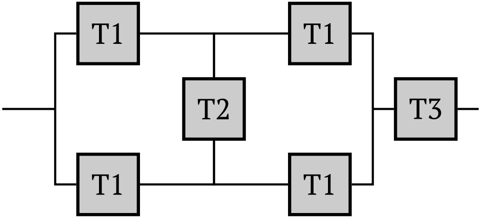
**Note:** this is a homogeneous polynomial of degree $K + 1$ in the survival signature and component survival probabilities $\implies$ can evaluate encrypted.

# Propagating uncertainty as a Bayesian

$$
\begin{aligned}
&P(T_{S^*} > t \mid \underline{y}_1, \dots \underline{y}_K) \\
&= \int \cdots \int P(T_{S^*} > t \mid p_t^1, \dots p_t^K) P(dp_t^1 \mid \underline{y}_1) \dots P(dp_t^K \mid \underline{y}_K) \\
&= \int \cdots \int \left[ \sum_{l_1=0}^{M_1} \cdots \sum_{l_K=0}^{M_K} \Phi(l_1, \dots, l_K) P\left( \bigcap_{k=1}^{K} \{C_t^k = l_k \mid p_t^k\} \right) \right] \\
&\qquad \times P(dp_t^1 \mid \underline{y}_1) \dots P(dp_t^K \mid \underline{y}_K) \\
&= \sum_{l_1=0}^{M_1} \cdots \sum_{l_K=0}^{M_K} \Phi(l_1, \dots, l_K) \prod_{k=1}^{K} \int P(C_t^k = l_k \mid p_t^k) P(dp_t^k \mid \underline{y}_k)
\end{aligned}
$$

A homogeneous polynomial of degree $K+1$ in the survival signature and posterior predictive component survival probabilities at each time point $\implies$ can still evaluate encrypted.

# Survival signature toy example



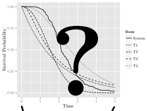

| T1 | T2 | T3 | Φ | T1 | T2 | T3 | Φ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 2 | 0 | 1 | 0.33 | 2 | 1 | 1 | 0.67 |
| 3 | 0 | 1 | 1 | 3 | 1 | 1 | 1 |
| 4 | 0 | 1 | 1 | 4 | 1 | 1 | 1 |

Table 1: Survival signature for a bridge system, omitting all rows with T3 $= 0$, since $\Phi = 0$ for these.

$k_s$ $k_p$

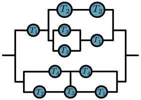
System Designer

$\nu$




$\underline{t} = \{t_1, \ldots, t_T\}$

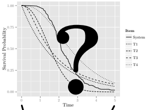
$\Phi(l_1, \ldots, l_K)$



$k_s$ $k_p$

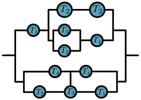
System Designer

$\nu$




$\underline{t} = \{t_1, \ldots, t_T\}$

$\Phi(l_1, \ldots, l_K)$

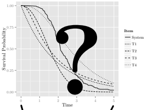


$$\Xi = \begin{pmatrix} 0 & \cdots & 0 & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(0, \ldots, 0) \rceil\right) \\ 0 & \cdots & 1 & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(0, \ldots, 1) \rceil\right) \\ & \vdots & & \vdots \\ l_1 & \cdots & l_K & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(l_1, \ldots, l_K) \rceil\right) \\ & \vdots & & \vdots \\ m_1 & \cdots & m_K & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(m_1, \ldots, m_K) \rceil\right) \end{pmatrix}$$

$k_s$ $k_p$

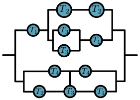
System Designer

$\nu$




$\underline{t} = \{t_1, \dots, t_T\}$

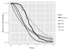
$\Phi(l_1, \dots, l_K)$



$$\Xi = \begin{pmatrix} 0 & \cdots & 0 & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(0, \dots, 0) \rceil\right) \\ 0 & \cdots & 1 & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(0, \dots, 1) \rceil\right) \\ & \vdots & & \vdots \\ l_1 & \cdots & l_K & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(l_1, \dots, l_K) \rceil\right) \\ & \vdots & & \vdots \\ m_1 & \cdots & m_K & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(m_1, \dots, m_K) \rceil\right) \end{pmatrix}$$


Manufacturer 1

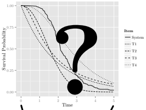
$\rightarrow$ $\underline{t}^1 \triangleq \{t_1^1, \dots, t_{n_1}^1\}$ $\rightarrow$



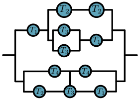
$k_s$ $k_p$

System Designer

$\nu$




$\Phi(l_1, \dots, l_K)$

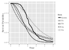


$$\Xi = \begin{pmatrix} 0 & \cdots & 0 & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(0,\dots,0) \rceil\right) \\ 0 & \cdots & 1 & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(0,\dots,1) \rceil\right) \\ & \vdots & & \vdots \\ l_1 & \cdots & l_K & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(l_1,\dots,l_K) \rceil\right) \\ & \vdots & & \vdots \\ m_1 & \cdots & m_K & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(m_1,\dots,m_K) \rceil\right) \end{pmatrix}$$

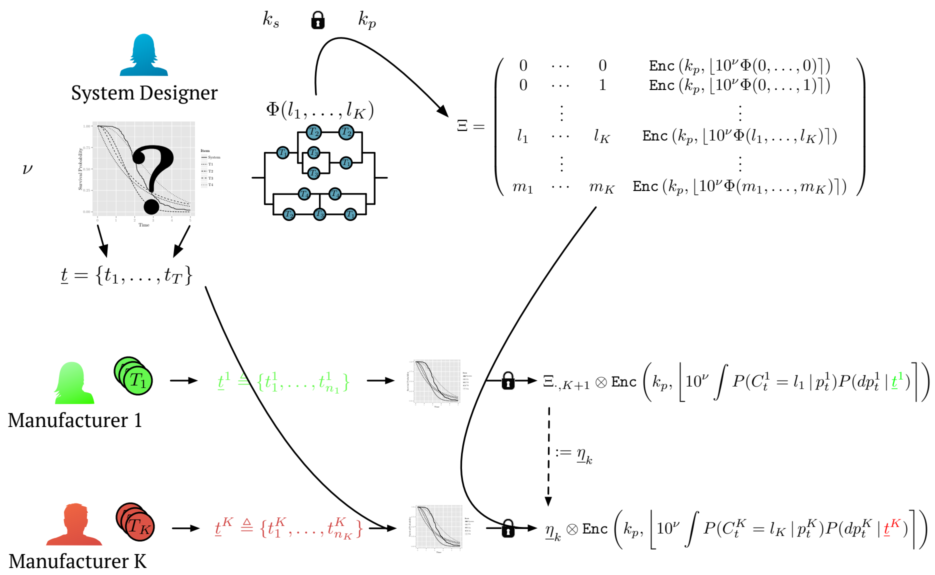
$\underline{t} = \{t_1, \dots, t_T\}$

Manufacturer 1

$\underline{t}^1 \triangleq \{t^1_1, \dots, t^1_{n_1}\}$

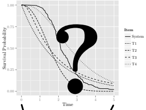


$$\Xi_{\cdot, K+1} \otimes \texttt{Enc}\left(k_p, \left\lfloor 10^\nu \int P(C^1_t = l_1 \mid p^1_t) P(dp^1_t \mid \underline{t}^1) \right\rceil \right)$$


$k_s$
$k_p$
System Designer
$\nu$
$\Phi(l_1, \ldots, l_K)$
$\Xi = \begin{pmatrix} 0 & \cdots & 0 & \texttt{Enc}\,(k_p, \lfloor 10^\nu \Phi(0,\ldots,0) \rceil) \\ 0 & \cdots & 1 & \texttt{Enc}\,(k_p, \lfloor 10^\nu \Phi(0,\ldots,1) \rceil) \\ & \vdots & & \vdots \\ l_1 & \cdots & l_K & \texttt{Enc}\,(k_p, \lfloor 10^\nu \Phi(l_1,\ldots,l_K) \rceil) \\ & \vdots & & \vdots \\ m_1 & \cdots & m_K & \texttt{Enc}\,(k_p, \lfloor 10^\nu \Phi(m_1,\ldots,m_K) \rceil) \end{pmatrix}$
$\underline{t} = \{t_1, \ldots, t_T\}$
Manufacturer 1
$\underline{t}^1 \triangleq \{t^1_1, \ldots, t^1_{n_1}\}$
$\Xi_{\cdot, K+1} \otimes \texttt{Enc}\left(k_p, \left\lfloor 10^\nu \int P(C^1_t = l_1 \mid p^1_t) P(dp^1_t \mid \underline{t}^1) \right\rceil \right)$
$:= \underline{\eta}_k$
Manufacturer K
$\underline{t}^K \triangleq \{t^K_1, \ldots, t^K_{n_K}\}$
$\underline{\eta}_k \otimes \texttt{Enc}\left(k_p, \left\lfloor 10^\nu \int P(C^K_t = l_K \mid p^K_t) P(dp^K_t \mid \underline{t}^K) \right\rceil \right)$

System Designer

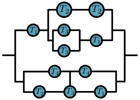
$\nu$




$\underline{t} = \{t_1, \dots, t_T\}$

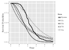
$k_s$ $k_p$

$\Phi(l_1, \dots, l_K)$

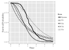


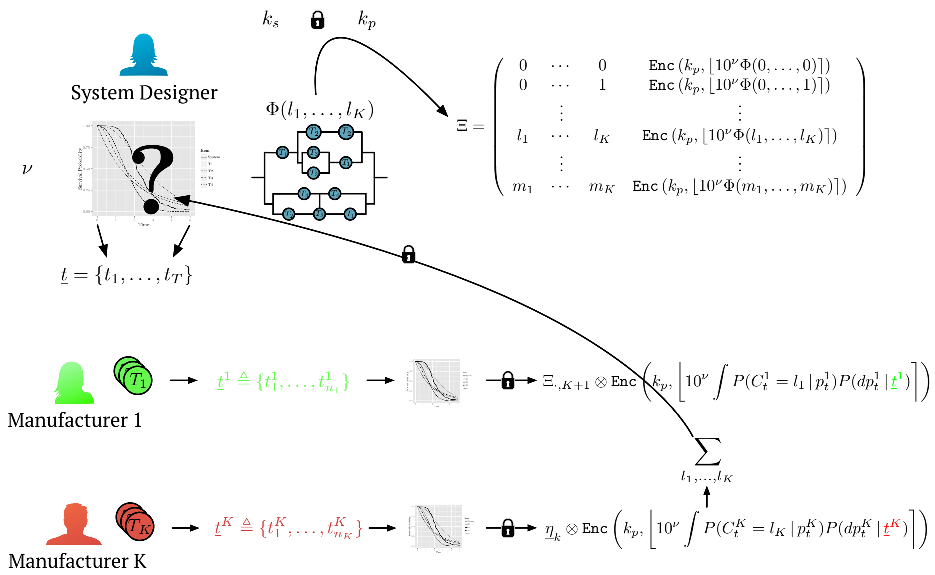

$$\Xi = \begin{pmatrix} 0 & \cdots & 0 & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(0,\dots,0)\rceil\right) \\ 0 & \cdots & 1 & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(0,\dots,1)\rceil\right) \\ & \vdots & & \vdots \\ l_1 & \cdots & l_K & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(l_1,\dots,l_K)\rceil\right) \\ & \vdots & & \vdots \\ m_1 & \cdots & m_K & \texttt{Enc}\left(k_p, \lfloor 10^\nu \Phi(m_1,\dots,m_K)\rceil\right) \end{pmatrix}$$

Manufacturer 1

$\underline{t}^1 \triangleq \{t^1_1, \dots, t^1_{n_1}\}$ →  → $\Xi_{\cdot,K+1} \otimes \texttt{Enc}\left(k_p, \left\lfloor 10^\nu \int P(C^1_t = l_1 \mid p^1_t) P(dp^1_t \mid \underline{t}^1)\right\rceil\right)$

Manufacturer K

$\underline{t}^K \triangleq \{t^K_1, \dots, t^K_{n_K}\}$ →  → $\underline{\eta}_k \otimes \texttt{Enc}\left(k_p, \left\lfloor 10^\nu \int P(C^K_t = l_K \mid p^K_t) P(dp^K_t \mid \underline{t}^K)\right\rceil\right)$


$k_s$
$k_p$
System Designer
$\nu$
$\Phi(l_1, \ldots, l_K)$
$\Xi = \begin{pmatrix} 0 & \cdots & 0 & \texttt{Enc}(k_p, \lfloor 10^\nu \Phi(0,\ldots,0) \rceil) \\ 0 & \cdots & 1 & \texttt{Enc}(k_p, \lfloor 10^\nu \Phi(0,\ldots,1) \rceil) \\ & \vdots & & \vdots \\ l_1 & \cdots & l_K & \texttt{Enc}(k_p, \lfloor 10^\nu \Phi(l_1,\ldots,l_K) \rceil) \\ & \vdots & & \vdots \\ m_1 & \cdots & m_K & \texttt{Enc}(k_p, \lfloor 10^\nu \Phi(m_1,\ldots,m_K) \rceil) \end{pmatrix}$
$\underline{t} = \{t_1, \ldots, t_T\}$
Manufacturer 1
$\underline{t}^1 \triangleq \{t_1^1, \ldots, t_{n_1}^1\}$
$\Xi_{\cdot,K+1} \otimes \texttt{Enc}\left(k_p, \left\lfloor 10^\nu \int P(C_t^1 = l_1 \mid p_t^1) P(dp_t^1 \mid \underline{t}^1) \right\rceil\right)$
$\sum_{l_1,\ldots,l_K}$
Manufacturer K
$\underline{t}^K \triangleq \{t_1^K, \ldots, t_{n_K}^K\}$
$\underline{\eta}_k \otimes \texttt{Enc}\left(k_p, \left\lfloor 10^\nu \int P(C_t^K = l_K \mid p_t^K) P(dp_t^K \mid \underline{t}^K) \right\rceil\right)$

# Example

# Example system

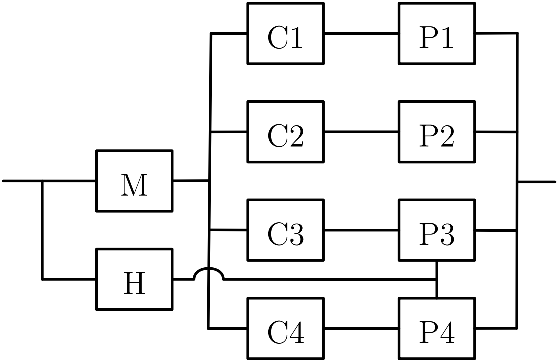



Figure 1: Simple automotive braking system. The master brake cylinder (M) engages all the four wheel brake cylinders (C1 – C4). These in turn each trigger a braking pad assembly (P1 – P4). The hand brake (H) goes directly to the rear brake pad assemblies P3 and P4; the vehicle brakes when at least one of the brake pad assemblies is engaged.

# Experimental results

In order to examine the practicality of the problem, perform a full encrypted analysis using Amazon EC2 cloud computing service to mimic a global supply chain.

| Role | Physical Server Location | Server Type |
|---|---|---|
| System designer | Dublin, Ireland | m4.10xlarge |
| Manufacturer C | Northern California, USA | m4.10xlarge |
| Manufacturer H | São Paulo, Brazil | c3.8xlarge |
| Manufacturer M | Sydney, Australia | r3.4xlarge |
| Manufacturer P | Tokyo, Japan | i2.8xlarge |

Precision was set to $\nu = 5$ and system designer specifies an evenly spaced time grid of 100 points $t \in [0, 5]$.

# Computational cost (I)

| Role | Action | Timing / Size |
|---|---|---|
| System designer Dublin, Ireland | Generation of $(k_p, k_s)$ | 0.3 secs |
| | Encryption of $\Xi^{(\Phi)}$ | 1 min 41.1 secs |
| | Saving $\Xi^{(\Phi)}$ to disk | 2 min 41.3 secs |
| | Compressing $\Xi^{(\Phi)}$ on disk | 48.0 secs |
| | Size of $\Xi^{(\Phi)}$ on disk | 5.5GB |

# Computational cost (I)

| Role | Action | Timing / Size | |
|---|---|---|---|
| System designer Dublin, Ireland | Generation of $(k_p, k_s)$ | | 0.3 secs |
| | Encryption of $\Xi^{(\Phi)}$ | 1 min | 41.1 secs |
| | Saving $\Xi^{(\Phi)}$ to disk | 2 min | 41.3 secs |
| | Compressing $\Xi^{(\Phi)}$ on disk | | 48.0 secs |
| | Size of $\Xi^{(\Phi)}$ on disk | 5.5GB | |
| *Transfer $\Xi^{(\Phi)}$ to Manufacturer C* | | 11 min | 37.5 secs |
| Manufacturer C Northern California, USA | | | |

# Computational cost (I)

| Role | Action | Timing / Size | |
|---|---|---|---|
| System designer Dublin, Ireland | Generation of $(k_p, k_s)$ | | 0.3 secs |
| | Encryption of $\Xi^{(\Phi)}$ | 1 min | 41.1 secs |
| | Saving $\Xi^{(\Phi)}$ to disk | 2 min | 41.3 secs |
| | Compressing $\Xi^{(\Phi)}$ on disk | | 48.0 secs |
| | Size of $\Xi^{(\Phi)}$ on disk | 5.5GB | |
| *Transfer $\Xi^{(\Phi)}$ to Manufacturer C* | | 11 min | 37.5 secs |
| Manufacturer C Northern California, USA | Decompress & load $\Xi^{(\Phi)}$ from disk | 10 min | 22.4 secs |
| | Update $\Xi^{(\Phi)}$ | 6 min | 18.3 secs |
| | Saving & compressing $\Xi^{(\Phi)}$ to disk | 2 min | 9.8 secs |

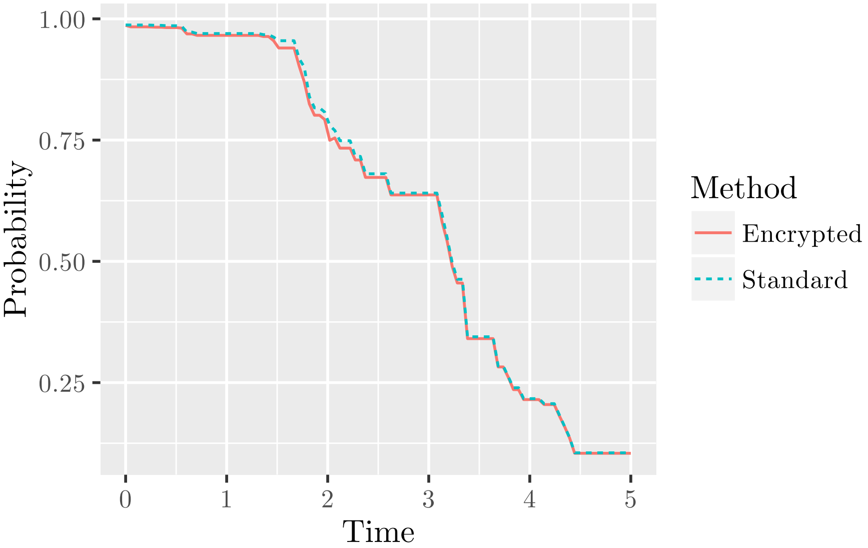
# Computational cost (I)

| Role | Action | Timing / Size | |
|---|---|---|---|
| System designer Dublin, Ireland | Generation of $(k_p, k_s)$ | | 0.3 secs |
| | Encryption of $\Xi^{(\Phi)}$ | 1 min | 41.1 secs |
| | Saving $\Xi^{(\Phi)}$ to disk | 2 min | 41.3 secs |
| | Compressing $\Xi^{(\Phi)}$ on disk | | 48.0 secs |
| | Size of $\Xi^{(\Phi)}$ on disk | 5.5GB | |
| *Transfer $\Xi^{(\Phi)}$ to Manufacturer C* | | 11 min | 37.5 secs |
| Manufacturer C Northern California, USA | Decompress & load $\Xi^{(\Phi)}$ from disk | 10 min | 22.4 secs |
| | Update $\Xi^{(\Phi)}$ | 6 min | 18.3 secs |
| | Saving & compressing $\Xi^{(\Phi)}$ to disk | 2 min | 9.8 secs |
| *Transfer $\Xi^{(\Phi)}$ to Manufacturer H* | | 11 min | 24.4 secs |
| Manufacturer H São Paulo, Brazil | Decompress & load $\Xi^{(\Phi)}$ from disk | 10 min | 13.2 secs |
| | Update $\Xi^{(\Phi)}$ | 7 min | 23.1 secs |
| | Saving & compressing $\Xi^{(\Phi)}$ to disk | 4 min | 45.2 secs |
| *Transfer $\Xi^{(\Phi)}$ to Manufacturer M* | | 20 min | 16.5 secs |
| Manufacturer M Sydney, Australia | Decompress & load $\Xi^{(\Phi)}$ from disk | 9 min | 41.0 secs |
| | Update $\Xi^{(\Phi)}$ | 11 min | 28.2 secs |
| | Saving & compressing $\Xi^{(\Phi)}$ to disk | 2 min | 54.2 secs |

# Computational cost (II)

| Role | Action | Timing / Size | | |
|---|---|---|---|---|
| | *Transfer $\Xi^{(\Phi)}$ to Manufacturer P* | | 6 min | 40.7 secs |
| Manufacturer P<br>Tokyo, Japan | Decompress & load $\Xi^{(\Phi)}$ from disk | | 9 min | 57.1 secs |
| | Update $\Xi^{(\Phi)}$ | | 7 min | 13.5 secs |
| | Compute $\xi$ | | | 6.1 secs |
| | Saving & compressing $\xi$ to disk | | | 2.5 secs |
| | Size of $\xi$ on disk | | 58.4MB | |
| | *Transfer $\xi$ to System Designer* | | | 39.5 secs |
| System designer<br>Dublin, Ireland | Decompress & load $\xi$ from disk | | | 5.9 secs |
| | Decryption of $\xi$ | | | 8.6 secs |
| **Total:** | | 2 hr | 18 min | 38.4 secs |

# Result


1.00
0.75
0.50
0.25
Probability
0
1
2
3
4
5
Time
Method
Encrypted
Standard

# R package

# HomomorphicEncryption R package (Aslett 2014)

```
library("HomomorphicEncryption")
p <- parsHelp("FandV", lambda=128, L=5)
k <- keygen(p)
c1 <- enc(k$pk, 2); c2 <- enc(k$pk, 3)
cres <- c1 + c2 * c1
dec(k$sk, cres)
```

```
[1] 8
```

```
cmat <- enc(k$pk, matrix(1:9, nrow=3))
cmat2 <- cmat %*% cmat
dec(k$sk, cmat2)
```

```
     [,1] [,2] [,3]
[1,]   30   66  102
[2,]   36   81  126
[3,]   42   96  150
```

# References

Aslett, L. J. M. (2014). HomomorphicEncryption: Fully homomorphic encryption. *R package version 0.1*.

Aslett, L. J. M., Coolen, F. P. A., & Wilson, S. P. (2015). Bayesian inference for reliability of systems and networks using the survival signature. *Risk Analysis*, 35/9: 1640–51. DOI: 10.1111/risa.12228

Aslett, L. J. M., Esperança, P. M., & Holmes, C. C. (2015). *A review of homomorphic encryption and software tools for encrypted statistical machine learning*. University of Oxford. Retrieved from <http://arxiv.org/abs/1508.06574>

Coolen, F. P. A., & Coolen-Maturi, T. (2012). Generalizing the signature to systems with multiple types of components. *Complex systems and dependability*, pp. 115–30. Springer.

Gentry, C. (2009). *A fully homomorphic encryption scheme* (PhD thesis). Stanford University. Retrieved from <crypto.stanford.edu/craig>

Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. *Foundations of Secure Computation*, 4/11: 169–80.