Abstract: More and more effort is being spent on security improvements in today’s computer networking environments. However, due to the nature of computer security, there is still a lack of good quantitative assessment methods. Inventing and developing new ways of measuring security are therefore needed in order to more exactly describe, assess, and improve the security of computer environments. One existing quantitative security measure is guesswork. Guesswork gives the average number of guesses in a brute force attack to succeed in breaking an encrypted message. In the current definition of guesswork, it is assumed that the attacker uses a single processor when breaking an encrypted message. An intelligent and motivated attacker will however likely use several processors that run in parallel. This paper formally investigates how guesswork changes over time in multi-processor attacks. The result is applied to three probability distributions, the English alphabet, the geometric and the truncated geometric, in order to illustrate some behaviors.

Keywords: computer security, security measures, guesswork, quantitative assessment, multi-processor attacks.

I. Introduction

The need for computer security in today’s open computer networks is undisputed. More and more effort is therefore being spent on security enhancing techniques to withstand attacks. However, due to the nature of computer security, there is still a lack of good quantitative assessment methods [5–7, 9, 11, 22], with the possible exception of measurement methods that are based on experiences, such as the common criteria [2]. Hence, it is necessary to invent and develop new ways of measuring security in order to more exactly describe, assess, and improve the security of computer environments. To address the issue of measuring computer security, we present and investigate how the probabilistic measure of guesswork changes over time in multi-processor attacks, that is, how guesswork changes through the number of incorrect guesses and the number of processors that the attacker has at his or her disposal. Neither time nor the number of processors in multi-processor attacks has to our knowledge previously been investigated for guesswork.

Guesswork [16, 18] is a quantitative security measure that gives the average number of guesses in an optimal brute force attack when breaking an encrypted message. Another interpretation of guesswork was given in [12], which states that guesswork gives the average number of guesses in an optimal linear search attack. In the same paper, guesswork and entropy [20], which is still the most commonly used probabilistic security measure, were proved to be connected through relative entropy [3].

Other means of measuring security that have an explicit or implicit time scale have also been proposed. A game theoretical approach to analyzing security of computer networks is used in [14]. An attempt to quantify security using game theory for modeling and computing the probabilities in a state transition diagram of expected attack behaviors is described in [19]. Moreover, quantifying operational security using state transition diagrams to model attacks and system restoration has been proposed in [8, 15, 21]. Work on developing quantitative metrics for network security monitoring and evaluation is also presented in [4], and work on security metrics in risk based methods is discussed and investigated in [10, 17].

The formal definition of guesswork, as given in Section II, only considers an attacker that has a single processor when conducting an attack. Today, a pool of computing resources, for instance in a cloud, might be used when breaking an encrypted message. Thus the attacker will be able to use several processors that can work in parallel to break an encrypted message. Furthermore, even if the attacker has only a single computer, it is highly likely that the computer is equipped with several processors or kernels. There is thus a need to formally investigate how guesswork changes with the number of processors in multi-processor attacks. It is also important to know and understand how well a system withstands an attack over time, i.e., how the guesswork of the system changes over time when the attacker guesses incorrect values. This paper combines both concepts by formally deriving expressions and investigating how guesswork changes over time because of the number of incorrect guesses in multi-processor attacks. The result is applied to three probability distributions, the English alphabet, the geometric and the truncated geometric, in order to illustrate some behaviors.
The remainder of the paper is organized as follows. The formal definition of guesswork is presented in Section II. Section III derives expressions for guesswork changes over time in a single-processor attack, dual-processor attack, and, finally, multi-processor attack. In Section IV, the derived results are used on three different probability distributions to illustrate some behaviors. The guesswork increment over time through the number of incorrect guesses is formally investigated in Section V. Finally, Section VI provides concluding remarks and future work.

II. A Note on Guesswork

Let \( \mathcal{X} = \{x_1, \ldots, x_n\} \) be a finite sample space, then the mapping

\[
p : \mathcal{X} \to [0, 1]
\]

is called a probability distribution if

\[
\sum_{i=1}^{n} p(x_i) = 1
\]

(2)

A \( \mathcal{X} \)-valued discrete random variable \( X \) with probability distribution \( p \) is a variable that attains values \( x_i \in \mathcal{X} \) with probability \( p(x_i) = p(X = x_i) \). If it is important from the context to point out that the random variable \( X \) is connected to the probability distribution \( p \), \( X_n \) is used. Furthermore, if \( A \subseteq \mathcal{X} \) and \( B \subseteq \mathcal{X} \) then \( p(A) \) is the marginal probability of \( A \), \( p(A|B) \) the conditional probability of \( A \) given \( B \), and \( p(A \cap B) \) the joint probability of \( A \) and \( B \). The three notations are related through the equation

\[
p(A \cup B) = p(A)p(B|A)
\]

(3)

Moreover, if \( A \) and \( B \) are mutually exclusive, \( A \cap B = \emptyset \), then the probability of \( A \) or \( B \) is

\[
p(A \cup B) = p(A) + p(B)
\]

(4)

Finally, if \( B = \hat{A} \), where \( \hat{A} = \mathcal{X} \setminus A \) is the complement of \( A \), then according to (2) and (4)

\[
p(A) + p(\hat{A}) = 1
\]

(5)

In an optimal brute force attack, the average number of guesses needed to find the value of \( X \) is called guesswork, \( W(X) \) [16,18]. When performing such an attack, the attacker is assumed to have complete knowledge of the probability distribution, \( p \), of \( X \). Hence, before the guessing process starts, the attacker arranges \( p \) in a non-increasing probability order

\[
p(x_1) \geq p(x_2) \geq \ldots \geq p(x_n)
\]

(6)

From (6), guesswork is defined as follows.

Definition 1. Given a probability distribution, \( p \), that is ordered according to (6), then guesswork \( W(X) \) is defined by

\[
W(X) = \sum_{i=1}^{n} ip(x_i)
\]

(7)

Hence, guesswork is the average of non-increasing ordered probability distributions. Moreover, in order to simplify and shorten the forthcoming derivations, the definition of guesswork is rewritten as

\[
W(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} p(x_j)
\]

(8)

where

\[
p(X_i) = p \left( \bigcup_{j=1}^{n} x_j \right)
\]

(9)

III. Guesswork Changes

This section derives expressions for changes in guesswork over time in a single-processor, dual-processor, and multi-processor attack, respectively.

A. Single-Processor Attack

To investigate how guesswork changes over time in a single-processor attack, it must be considered how guesswork changes during the guessing attack when the attacker guesses incorrect values according to (6). The most probable value that the attacker first will guess is \( x_1 \). If this value is incorrect, guesswork changes from \( W(X) = W(X|X_1) \) to

\[
W(X|X_2) = \sum_{i=2}^{n} p(X_i|X_2)
\]

(10)

Accordingly, the second most probable value that the attacker will guess is \( x_2 \). If this value is also incorrect, guesswork changes from \( W(X|X_2) \) to

\[
W(X|X_3) = \sum_{i=3}^{n} p(X_i|X_3)
\]

(11)

Generalizing, after \( k \) incorrect guesses, the expression for guesswork in a single-processor attack becomes

\[
W(X|X_{k+1}) = \sum_{i=k+1}^{n} p(X_i|X_{k+1})
\]

(12)

Finally, it is also possible to express \( W(X|X_{k+1}) \) in terms of
\[ W(X|A_k) \] by observing that
\[
W(X|A_{k+1}) = \frac{1}{p(A_{k+1})} \left( \sum_{i=k}^{n} p(X_i) - p(X_k) \right) \\
= \frac{p(X_k)}{p(A_{k+1})} (W(X|A_k) - 1) \tag{13}
\]

### B. Dual-Processor Attack

In an optimal brute force attack with two processors, the attacker splits (6) into two probability distributions, \( p(x_{2i-1}|A_1) \) and \( p(x_{2i}|A_2) \), where
\[
A_1 = \{ x_{2i-1} | 1 \leq i \leq \left\lceil \frac{n}{2} \right\rceil \} \tag{14}
\]
and
\[
A_2 = \{ x_{2i} | 1 \leq i \leq \left\lceil \frac{n-1}{2} \right\rceil \} \tag{15}
\]

Hence, the attacker feeds one processor with values of odd index numbers, \( A_1 \), in a non-increasing probability order and the other processor with values of even index numbers, \( A_2 \), in a non-increasing probability order. Note that \( \chi = A_1 \cup A_2 \). From (14) the guesswork of the first processor, i.e., odd indexes, becomes
\[
W(X|A_1) = \sum_{i=1}^{|A_1|} p(X_{2i-1}|A_1) \\
= \frac{1}{p(A_1)} \sum_{i=1}^{|A_1|} p(\hat{A}_1 \cap X_{2i-1}) \tag{16}
\]
and from (15) the guesswork of the second processor, i.e., even indexes, becomes
\[
W(X|A_2) = \sum_{i=1}^{|A_2|} p(X_{2i}|A_2) \\
= \frac{1}{p(A_2)} \sum_{i=1}^{|A_2|} p(\hat{A}_2 \cap X_{2i}) \tag{17}
\]

The average guesswork of (16) and (17) gives the guesswork in a dual-processor attack. Hence,
\[
W_2(X) = \frac{1}{2} p(A_j) W(X|A_j) \\
= \frac{1}{2} \sum_{j=1}^{|A_j|} \sum_{i=1}^{|A_j|} p(A_j \cap X_{2(i-1)+j}) \\
= \sum_{i=1}^{|A_1|} p(X_{2i-1}) \tag{18}
\]

where subscript in \( W_2(X) \) has been used to denote the number of processors used in the attack.

Another way of calculating \( W_2(X) \), suggested by row three in (18), is to create the probability distribution, \( \hat{p} \), by summing two consecutive probabilities in \( p \) as
\[
\hat{p}(y_i) = p(x_{2i-1}) + p(x_{2i}) \tag{19}
\]
giving
\[
\hat{p}(Y_i) = p(X_{2i-1}) \tag{20}
\]
and then use the following observation
\[
W_2(X_p) = \sum_{i=1}^{|A_1|} p(X_{2i-1}) \\
= \sum_{i=2}^{|A_1|} \hat{p}(Y_i) \\
= W(X_p) \tag{21}
\]

Hence, calculating the guesswork of \( p \) using two processors gives the same result as calculating the guesswork of \( \hat{p} \) using one processor. Note that since \( p \) is arranged in a non-increasing probability order so is \( \hat{p} \).

To investigate how guesswork changes over time in a dual-processor attack, it must be considered how guesswork changes during the guessing attack when the attacker guesses two incorrect values at the same time according to (6), called a 2 guessing round. The two most probable values that the attacker will guess in the first 2 guessing round guess are \( x_1 \) and \( x_2 \). If both values are incorrect, and by assuming that the two processors have the same processing speed, the guesswork at a single point in time changes from
\[
W_2(X_p) = W_2(X_p|X_1) \\
W_2(X_p|X_3) = W(X_p|Y_2) \\
W_2(X_p|X_{2k+1}) = W(X_p|Y_{k+1}) \\
W_2(X_p|X_{2k+1}) = \sum_{i=1}^{|A_1|} p(X_{2i-1}) \tag{23}
\]

Generalizing, after \( k \) incorrect 2 guessing rounds, the expression for guesswork in a dual-processor attack becomes
\[
W_2(X_p|X_{2k+1}) = W(X_p|Y_{k+1}) \\
= \sum_{i=1}^{|A_1|} p(X_{2i-1}) \tag{24}
\]

where \( 0 \leq k < |A_1| \).

Finally, as in the single-processor case, by using (13) and (21), \( W_2(X_p|X_{2k+1}) \) can be expressed in terms of \( W_2(X_p|X_{2k-1}) \) as
\[
W_2(X_p|X_{2k+1}) = W(X_p|Y_{k+1}) \\
= \frac{\hat{p}(Y_{k+1})}{p(X_{k+1})} (W(X_p|Y_k) - 1) \\
= \frac{p(X_{k+1})}{p(X_{2k+1})} (W_2(X_p|X_{2k-1}) - 1) \tag{24}
\]

### C. Multi-Processor Attack

In an optimal brute force attack with \( m \) processors, the attacker splits (6) into \( m \) probability distributions,
Another way of calculating \( p(x_{m(i-1)+j} | A_j) \), \( 1 \leq j \leq m \), where

\[
A_j = \{ x_{m(i-1)+j} \mid 1 \leq i \leq \left\lfloor \frac{n+1-j}{m} \right\rfloor \}
\]  

(25)

Hence, the attacker feeds each of the \( m \) processors with values of a multiple of \( m \) of the index values in a non-increasing probability order. Note that \( \chi = \bigcup_{j=1}^{m} A_j \).

From (25) the guesswork of each processor becomes

\[
W(X|A_j) = \frac{1}{p(A_j)} \sum_{i=1}^{|A_j|} p(A_j \cap \chi_{m(i-1)+j})
\]  

(26)

The average guesswork of (26) gives guesswork in a multi-processor attack. Hence,

\[
W_m(X) = \sum_{j=1}^{m} \frac{1}{p(A_j)} \sum_{i=1}^{|A_j|} p(A_j \cap \chi_{m(i-1)+j})
\]

(27)

Another way of calculating \( W_m(X) \), suggested by row three in (27), is to create probability distribution \( \hat{p} \) by summing \( m \) consecutive probabilities in \( p \) as

\[
\hat{p}(y_i) = \sum_{j=1}^{m} p(x_{m(i-1)+j})
\]  

(28)

giving

\[
\hat{p}(Y_i) = p(\chi_{m(i-1)+1})
\]  

(29)

and then use the following observation

\[
W_m(X_p) = \sum_{|A_i| = 1} |A_i| p(\chi_{m(i-1)+1})
\]

(30)

Hence, calculating the guesswork of \( p \) using \( m \) processors gives the same result as calculating the guesswork of \( \hat{p} \) using a single processor. Note again that since \( p \) is arranged in a non-increasing probability order so is \( \hat{p} \).

To investigate how guesswork changes over time in a multi-processor attack, it must be considered how the guesswork changes during the guessing attack when the attacker guesses \( m \) incorrect values at the same time according to (6), called an \( m \) guessing round. The \( m \) most probable values that the attacker will guess in the first \( m \) guessing round guess are \( x_1, \ldots, x_m \). If all \( m \) values are incorrect, and by assuming that the \( m \) processors have the same processing speed, the guesswork at a single point in time changes from \( W_m(X_p) = W_m(X_p|X_{m+1}) \) to

\[
W_m(X_p|X_{m+1}) = W(X_p|Y_2)
\]

(31)

Generalizing, after \( k \) incorrect \( m \) guessing rounds, the expression for guesswork in a multi-processor attack becomes

\[
W_m(X_p|X_{mk+1}) = W(X_p|Y_{k+1})
\]

(32)

where \( 0 \leq k < |A_1| \).

Finally, as in the single-processor case, by using (13) and (30), \( W(X_p|X_{mk+1}) \) can be expressed in terms of \( W(X_p|X_{mk(k-1)+1}) \) as

\[
W_m(X_p|X_{mk+1}) = W(X_p|Y_{k+1})
\]

(33)

IV. Probability Distributions

In this Section, the English alphabet (Section IV-A), the geometric (Section IV-B), and the truncated geometric (Section IV-C) are used to illustrate some behaviors of the previously derived results. The geometric and truncated geometric distributions are used since they are both ordered in a non-increasing order and resemble in shape the English alphabet distribution.

A. The English Alphabet Distribution

The English alphabet probability distribution \( e[1] \), arranged in a non-increasing probability order, is plotted in Figure 1. Figure 2 illustrates the changes in guesswork of \( e \) in a single-processor attack (12) (upper plot) and in a dual-processor attack (23) (lower plot) when one letter is encrypted. Note that the two plots have more or less the same shape and that two processors almost halve guesswork. For example, \( W_2(X_e) = 7.536 \) while \( W_2(X_e) = 4.040 \), and \( W_2(X_e|X_{11}) \approx 5.5 \) while \( W_2(X_e|X_{11}) \approx 3. \) In Figure 2, the guesswork decreases after each incorrect guess, except after guess 22, letter ‘k’, in the single-processor case and after guessing round 11, letters ‘v’ and ‘k’, in the dual-processor case, where the guesswork locally increases. This is due to the fact that the probability distribution of \( e \) then becomes more uniform and makes the attacker more uncertain. Furthermore, the guesswork becomes one when \( k = 25 \) in the single-processor attack and when \( k = 12 \) in the dual-processor attack.
Figure. 1: The English alphabet probability distribution arranged in a non-increasing probability order.

Figure. 2: Guesswork changes of e in a single-processor attack (upper plot) and in a dual-processor attack (lower plot) when one letter is encrypted.

Instead of plotting the guesswork changes of e against the number of incorrect guesses as in Figure 2, the guesswork changes of e against the number of processors (32) when one letter is encrypted are illustrated in Figure 3. The eight plots have $k = 0, \ldots, 7$, ranging in increasing order from the highest to the lowest plot. Note that Figure 2 for $k = 0, \ldots, 7$ gives the corresponding guesswork when $m = 1$ and $m = 2$.

B. The Geometric Distribution

The geometric probability distribution, $g$, by using the complement $\delta = 1 - \delta$ as the generating probability is given by the probabilities

\[
g(x_i) = \delta(1 - \delta)^{i-1} = \delta \delta^{i-1}
\]  

(34)

Hence,

\[
g(x_i) = \sum_{j=i}^{\infty} \delta \delta^{j-1} = \delta^{i-1}
\]  

(35)

By using (35) in (8), the guesswork of $g$ in a single-processor attack becomes

\[
W(X_g) = \sum_{i=1}^{\infty} \delta^{i-1} = \frac{1}{1 - \delta}
\]  

(36)

Furthermore, the sequence of guesswork changes of $g$ in a single-processor attack, using (35) in (12), becomes

\[
W(X_g|X_{k+1}) = \sum_{i=k+1}^{\infty} \frac{\delta^{i-1}}{\delta^k} = \frac{1}{1 - \delta}
\]  

(37)

This sequence is independent with respect to $k$, which implies that the guesswork of $g$ remains constant during a single-processor attack. This comes from the fact that the geometric probability distribution is not transformed during an attack. That is, after each incorrect guess, the distribution is preserved in shape.

In a dual-processor attack, the guesswork of $g$ becomes

\[
W_2(X_g) = \sum_{i=1}^{\infty} \delta^{2(i-1)} = \frac{1}{1 - \delta^2}
\]  

(38)
and the sequence of guesswork changes
\[
W_2(X_g|X_{2k+1}) = \sum_{i=k+1}^{\infty} \frac{\delta^2(i-1)}{\delta^{2k}} = \frac{1}{1 - \delta^2} \tag{39}
\]

Generally, in a multi-processor attack, the guesswork of \( g \) becomes
\[
W_m(X_g) = \sum_{i=1}^{\infty} \delta^{m(i-1)} = \frac{1}{1 - \delta^m} \tag{40}
\]
and the sequence of guesswork changes
\[
W_m(X_g|X_{mk+1}) = \sum_{i=k+1}^{\infty} \frac{\delta^{m(i-1)}}{\delta^{mk}} = \frac{1}{1 - \delta^m} \tag{41}
\]

Note that the general sequence of guesswork changes is also independent with respect to \( k \). Hence, the guesswork of \( g \) remains constant during multi-processor attacks.

The English alphabet probability distribution, plotted in Figure 1, graphically resembles the geometric probability distribution. By using maximum likelihood estimation, \( \delta \) can be estimated by setting the mean of the observed English alphabet probability distribution equal to the mean of the estimated geometrical probability distribution. Hence, setting \( W(X_e) = W(X_g) \) gives that \( (1 - \delta)^{-1} = W_1(X_e) \approx 0.133 \), with an maximum error between the two corresponding cumulative distributions of 0.065. Figure 4 illustrates the probability distributions \( e \) (bars) and estimated \( g \) of \( e \) (solid line).

Furthermore, the sequence of guesswork changes of the estimated \( g \) when one letter is encrypted would in Figure 2 be plotted by a straight line having \( W(X_g|X_{k+1}) = 7.536 \) for the single-processor attack and \( W_2(X_g|X_{2k+1}) = 4.036 \) for the dual-processor attack. Thus, even if the two probability distributions are quite similar in nature, their sequence of guesswork changes behaves completely differently. Generally, a multi-processor attack would in Figure 2 be a straight line having
\[
W_m(X_g|X_{mk+1}) = \frac{1}{1 - (1 - \frac{1}{W_1(X_e)^m})} \tag{42}
\]

Note that even if (42) is independent of the number of incorrect guessing rounds, \( k \), it will decrease with a greater number of processors.

C. The Truncated Geometric Distribution

The geometric probability distribution, \( g_t \), has an infinite support. That is, it has an infinite sample space. Hence, as concerns a finite sample space, the truncated geometric probability distribution, \( g_t \), which has a finite support, might be a better choice. By again using the complement \( \hat{\delta} = 1 - \delta \) as the generating probability, the probabilities of \( g_t \) on \( 1 \leq i \leq n \) is given by
\[
g_t(x_i) = \frac{\hat{\delta}^{i-1}}{1 - \delta^n} \tag{43}
\]

Hence,
\[
g_t(X_i) = \sum_{j=1}^{n} \frac{\hat{\delta}^{j-1}}{1 - \delta^n} = \frac{\delta^{i-1} - \delta^n}{1 - \delta^n} \tag{44}
\]

By using (44) in (8), the guesswork of \( g_t \) in a single-processor attack becomes
\[
W(X_g) = \sum_{i=1}^{n} \frac{\hat{\delta}^{i-1} - \delta^n}{1 - \delta^n} = \frac{1}{1 - \delta} \frac{(n-k)\delta^n-k}{1 - \delta^n-k} \tag{45}
\]

Furthermore, the sequence of guesswork changes of \( g_t \) in a single-processor attack, using (44) in (12), becomes
\[
W(X_g|X_{k+1}) = \sum_{i=k+1}^{\infty} \frac{\hat{\delta}^{i-1} - \delta^n}{\delta^k - \delta^n} = \frac{1}{1 - \delta} \frac{(n-k)\delta^n-k}{1 - \delta^n-k} \tag{46}
\]

Note that (46) is equal to (45) when \( k = 0 \) and equal to one when \( k = n - 1 \). Moreover, (46) \( \to \) (37) when \( n \to \infty \).

In a dual-processor attack, the guesswork of \( g_t \) becomes
\[
W_2(X_g) = \sum_{i=1}^{\frac{n}{2}} \frac{\delta^2(i-1) - \delta^n}{1 - \delta^n} = \frac{1 - \delta^2[\frac{n}{2}]}{(1 - \delta^2)(1 - \delta^n)} = \frac{[\frac{n}{2}]\delta^n}{1 - \delta^n} \tag{47}
\]

and the sequence of guesswork changes
\[
W_2(X_g|X_{2k+1}) = \sum_{i=k+1}^{\frac{n}{2}} \frac{\delta^2(i-1) - \delta^n}{\delta^{2k} - \delta^n} = \frac{1 - \delta^2[\frac{n}{2}] \cdot 2k}{(1 - \delta^2)(1 - \delta^{n-2k})} = \frac{[\frac{n}{2}] - k}{1 - \delta^{n-2k}} \tag{48}
\]
Generally, in a multi-processor attack, the guesswork of $g_t$ becomes

$$W_m(X_{g_t}) = \sum_{i=1}^{\lceil \frac{n}{m} \rceil} \frac{\delta^{m(i-1)} - \delta^n}{1 - \delta^n} = \frac{1 - \delta^{m(\lceil \frac{n}{m} \rceil)}}{(1 - \delta^m)(1 - \delta^n)}$$

and the sequence of guesswork changes

$$W_m(X_{g_t}|X_{mk+1}) = \sum_{i=k+1}^{\lceil \frac{n}{m} \rceil} \frac{\delta^{m(i-1)} - \delta^n}{\delta^mk - \delta^n} = \frac{1 - \delta^{m(\lceil \frac{n}{m} \rceil) - k}}{(1 - \delta^m)(1 - \delta^n - mk)}$$

Note that (48) is equal to (47) and (50) is equal to (49) when $k = 0$, and is equal to one when $k = \lceil \frac{n}{m} \rceil - 1$ or $k = \lceil \frac{n}{m} \rceil - m$, respectively. Furthermore, (48) $\to$ (39) and (50) $\to$ (41) when $n \to \infty$.

As for the probability distribution, $g$, $\delta$ can be estimated by setting the mean of the observed English alphabet probability distribution to the mean of the estimated truncated geometrical probability distribution. Hence, solving the equation $W(X_e) = W(X_{g_t})$ gives that $1 - \delta = 0.116$, with a maximum error between the two corresponding cumulative distributions of 0.035. Thus, according to the maximum error, it is better to use the estimation $g_t$ of $e$ instead of the estimation $g$ of $e$. Figure 5 illustrates the probability distributions $e$ (bars), estimated $g$ of $e$ (dotted line), and estimated $g_t$ of $e$ (solid line). Figure 6 illustrates the guesswork changes of $e$ and the estimated $g_t$ of $e$ in a dual-processor attack (two lower plots) and in a single-processor attack (two upper plots) when one letter is encrypted. Moreover, Figure 7 illustrates the guesswork changes of $e$ and of the estimated $g_t$ of $e$ against the number of processors in a multi-processor attack when one letter is encrypted. The two upper plots have $k = 1$ and the two lower plots have $k = 4$, where $k$ gives the number of incorrect $m$ guessing rounds, with $e$ corresponding to the lowest plot in both cases.

The guesswork changes of $g_t$ for other values of $\delta = \frac{1}{2}$ in a single-processor attack (upper plot) and in a dual-processor attack (lower plot) are shown in Figure 8 when $\gamma \in \{1, \ldots, 20\}$ and $n = 50, 100$ and 200, respectively. Note that the corresponding $g$ of $g_t$ would be a straight line with a value of $\gamma^2$ in the single-processor case and a value of $\frac{\gamma^2}{2\gamma-1}$ in the dual-processor case.
V. Guesswork Increment

The guesswork increment is the change in or difference between two consecutive changes in guesswork. Thus, the guesswork increment regarding $k$ is given by

$$
\Delta W^k_m(X_p) = W_m(X_p|X_{m(k+1)}) - W_m(X_p|X_{mk+1})
= W(X_p|Y_{k+2}) - W(X_p|Y_{k+1})
= \frac{\hat{p}(Y_{k+1})}{\hat{p}(Y_{k+2})} (W(X_p|Y_{k+1}) - W(X_p|Y_{k+1} + 1))
= \frac{\hat{p}(Y_{k+1})}{\hat{p}(Y_{k+2})} (1 - \frac{1}{\hat{p}(Y_{k+1})}) W(X_p|Y_{k+1}) - 1)
$$

(51)

where $0 \leq k < |A_1| - 1$, and the probability distribution, $\hat{p}$, (28), have been used instead of $p$ to shorten the notation. The first factor in (51) is always positive, whereas the second factor can either be positive or negative. Hence, the guesswork decreases, $\Delta W^k_m(X) < 0$, if

$$
W(X_p|Y_{k+1}) < \frac{\hat{p}(Y_{k+1})}{\hat{p}(Y_{k+1}) - \hat{p}(Y_{k+2})}
$$

(52)

and increases, $\Delta W^k_m(X) > 0$, if

$$
W(X_p|Y_{k+1}) > \frac{\hat{p}(Y_{k+1})}{\hat{p}(Y_{k+1}) - \hat{p}(Y_{k+2})}
$$

(53)

where

$$
\frac{\hat{p}(Y_{k+1})}{\hat{p}(Y_{k+1}) - \hat{p}(Y_{k+2})} = \frac{1}{\hat{p}(y_{k+1}|Y_{k+1})}
$$

(54)

is the probability of the current most probable value. In Figure 6, the guesswork increment is positive after guess 21, letter ‘v’, in the single-processor case and after guess 10, letters ‘p’ and ‘b’, in the dual-processor case. Moreover, Figure 9 plots $\Delta W^k_m(X) = 0$, $\Delta W^k_m(X) < 0$ below the plot and $\Delta W^k_m(X) > 0$ above the plot. For instance, if $W(X_p) = 2$, then the guesswork decreases if $\hat{p}(y_1) < \frac{1}{2}$ and increases if $\hat{p}(y_1) > \frac{1}{2}$.

For the geometric probability distribution (34)

$$
\frac{\hat{g}(Y_{k+1})}{\hat{g}(Y_{k+2})} = \frac{\delta^{mk}}{\delta^{mk(k+1)}} = \frac{1}{1 - \delta^m}
$$

(55)

Comparing with (41) gives

$$
W(X_p|Y_{k+1}) = \frac{\hat{g}(Y_{k+1})}{\hat{g}(Y_{k+1}) - \hat{g}(Y_{k+2})}
$$

(56)

and, thus, $\Delta W^k_m(X) = 0$. This implies that the guesswork increment of the geometric probability distribution lies on the plot in Figure 9, and, hence, $W_m(X_p|Y_{mk+1})$ does not change its value during an attack.

For the truncated geometric probability distribution (43)

$$
\frac{\hat{g}(Y_{k+1})}{\hat{g}(Y_{k+2})} = \frac{\delta^{mk} - \delta}{\delta^{mk(k+1)} - \delta^{mk}} = \frac{1 - \delta^{-mk}}{1 - \delta^{-m}}
$$

(57)

By using the Taylor series

$$
\delta^x = 1 + x \ln(\delta) + \frac{x^2 \ln(\delta)^2}{2!} + \ldots
$$

(58)

and the relation

$$
(-n) \mod m = m \left\lceil \frac{n}{m} \right\rceil - n
$$

(59)

it is possible to show that

$$
W(X_p|Y_{k+1}) < \frac{1 - \delta^{-n-mk}}{1 - \delta^{-m}}
$$

(60)
Hence, $\Delta W^k_m(X_{g_t}) < 0$, giving that the truncated geometric probability distribution lies below the plot in Figure 9, and $W_m(X_{g_t}|X_{mk+1})$ decreases.

In Figure 10, the guesswork increment of $e$ (solid line) and estimated $g_t$ of $e$ (dotted line) in a single-processor attack (upper plot) and in a dual-processor attack (lower plot) are plotted when one letter is encrypted. Note the two spikes where the guesswork increment is positive, corresponding to the increase in the guesswork in Figure 6.

Another probability distribution that has not yet been mentioned in this paper is the uniform probability distribution, $u$, which maximizes guesswork. For this distribution

$$u(X_i) = \sum_{j=1}^{n} \frac{1}{n} = \frac{n-i+1}{n}$$  \hspace{1cm} (61)

Hence, the guesswork of $u$ in a multi-processor attack becomes

$$W_m(X_u) = \sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \frac{n-m(i-1)}{n} = \frac{n+m}{2m} + \frac{r^{-n}(m-r^{-n})}{2mn}$$  \hspace{1cm} (62)

and the sequence of guesswork changes

$$W_m(X_u|X_{mk+1}) = \sum_{i=mk+1}^{\lfloor \frac{n}{m} \rfloor} \frac{n-m(i-1)}{n-mk} = \frac{n-m(k-1)}{2m} + \frac{r^{-n}(m-r^{-n})}{2m(n-mk)}$$  \hspace{1cm} (63)

where $r^{-n} = (-n) \mod m$ from (59) has been used to shorten the notation. Note that (63) is equal to (62) when $k = 0$ and equal to one when $k = \lfloor \frac{n}{m} \rfloor - 1$. Moreover, if $m = 1$, then $W_m(X_u) = \frac{n+1}{2}$, which yields the previously known result of the guesswork of $u$.

By using the inequalities $k \leq \lfloor \frac{m}{n} \rfloor - 2$ and $r^{-n} < m$, the following inequality can be shown to hold

$$2m - (n-mk) \leq r^{-n} < n - mk$$  \hspace{1cm} (64)

Thus,

$$W_m(X_u|X_{mk+1}) < \frac{n-mk+m}{2m} + \frac{n-mk-m}{2m} = \frac{n-mk}{m}$$  \hspace{1cm} (65)

and since

$$\frac{\hat{u}(Y_{k+1})}{\hat{u}(Y_{k+2})} = \frac{n-mk}{m}$$  \hspace{1cm} (66)

$\Delta W^k_m(X_u) < 0$ and, hence, $W_m(X_u|X_{mk+1})$ decreases.

Figure 11 illustrates the guesswork changes of $u$ in a single-processor and a dual-processor attack when one letter is encrypted. In the figure, the plots from Figure 6 are also shown.

![Figure 10: Guesswork increment of e (solid line) and estimated g_t of e (dotted line) in a single-processor attack (upper plot) and in a dual-processor attack (lower plot) when one letter is encrypted.](image)

![Figure 11: Guesswork changes of u in a single- and dual-processor attack when one letter is encrypted. In the figure, the plots from Figure 6 are also shown.](image)

### VI. Concluding Remarks and Future Work

Since the current formal definition of guesswork considers only the case of an attacker that has a single processor when conducting an attack, we have in this paper formally investigated how guesswork changes over time through the number of incorrect guesses in single-processor, dual-processor, and multi-processor attacks, respectively. The derived result was used in three probability distributions, the English alphabet, the geometric, and the truncated geometric, in order to illustrate some behaviors.

The goal of our future work is to further investigate properties of guesswork changes in multi-processor attacks by investigating the relationship to joint and conditional guesswork as defined in [13]. We will also study how the guesswork increment behaves, that is, how the difference between
two consecutive values of the change in guesswork sequence behaves.

Another interesting investigation would be to see how entropy changes in a multi-processor attack and how guesswork and entropy are related during such attacks. The aim of such findings is to provide a better understanding of guesswork and entropy, and the work is thus a step towards building up a theory of probabilistic security measures.

Acknowledgment

The work has been carried out within the Compare Business Innovation Centre phase 3 (C-BIC 3) project, funded partly by the European Regional Development Fund (ERDF).

References


Author biographies

**Reine Lundin** received his Licentiate degree in Computer Science from Karlstad University, Sweden, in 2007. He also received a Master’s Degree in Physics and a Master’s degree in Mathematics from Karlstad University in 1999 and 2003, respectively. He joined the Department of Computer Science at Karlstad University in 2000, where he currently works as a lecturer. His research focus is quantitative security metrics and tunable security services. He has authored/coauthored over 25 book chapters and journal and conference papers.

**Stefan Lindskog** received his Licentiate and PhD degrees in Computer Engineering from Chalmers University of Technology, Göteborg, Sweden in 2000 and 2005, respectively. In 2008, he received the Docent degree in Computer Science at Karlstad University, Sweden. He joined the Department of Computer Science at Karlstad University in 1990, where he is currently a full professor. His research focus is the design of tunable and adaptable security services and security and performance analysis of security services and protocols. He has authored/coauthored one textbook, eight book chapters, and over 50 journal and conference papers.